

Technical Guide

GUIDE #2: A Data Replication Scenario using Ezstackr Cloud

Introduction:

High Availability (HA) and Disaster Recovery (DR) is one of most important functions in a data center. The Ezstackr cloud supports HA and DR for mission critical business. You can quickly recover from a disaster and ensure business continuity of your applications while keeping your costs down.

Disaster recovery is about preparing for and recovering from a disaster. Any event that has a negative impact on your business continuity or finances could be termed a disaster. This could be hardware or software failure, a network outage, a power outage, physical damage to a building like fire or flooding, human error, or some other significant disaster.

Meeting an organization's disaster recovery challenges requires addressing problems from several angles based on specific recovery point and recovery time objectives. Today's tight RTO and RPO expectations mean almost no data gets lost and no downtime. To meet those expectations, businesses must move beyond backup and consider a data replication strategy.

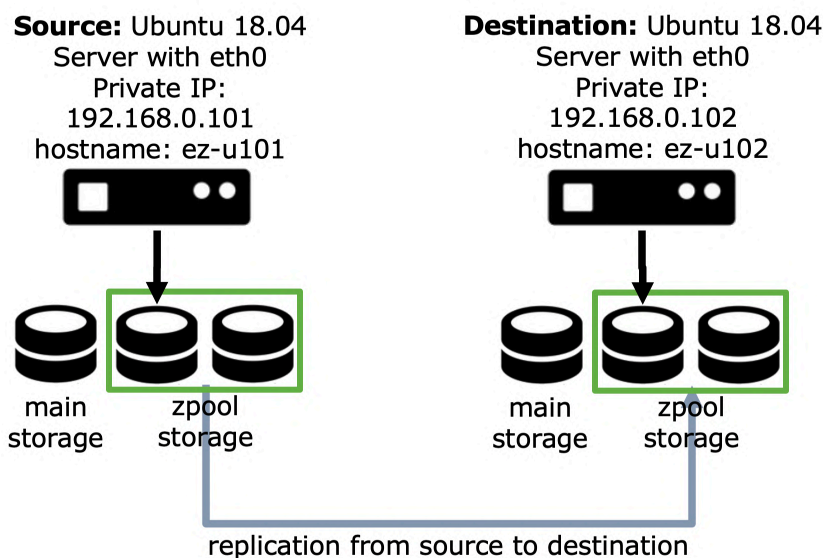
Replication Scenario Summary:

Here is a realistic data replication test scenario designed to help users design and implement their own DR solutions. In this test scenario, the **Source** will replicate data to the **Destination**.

1. Creating Source and Destination VM
2. Configuring Storage for Source and Destination VM
3. Configuring ZFS for Source and Destination VM
4. Configuring OPENSSH
5. Creating Snapshots on Source VM
6. Replicating Snapshots on Source VM to Destination VM
7. Replicating Snapshots on Source VM to Destination VM where Source and Destination Are In Separate Locations / Geography
8. Checking Integrity of Replicated Snapshots on Destination VM



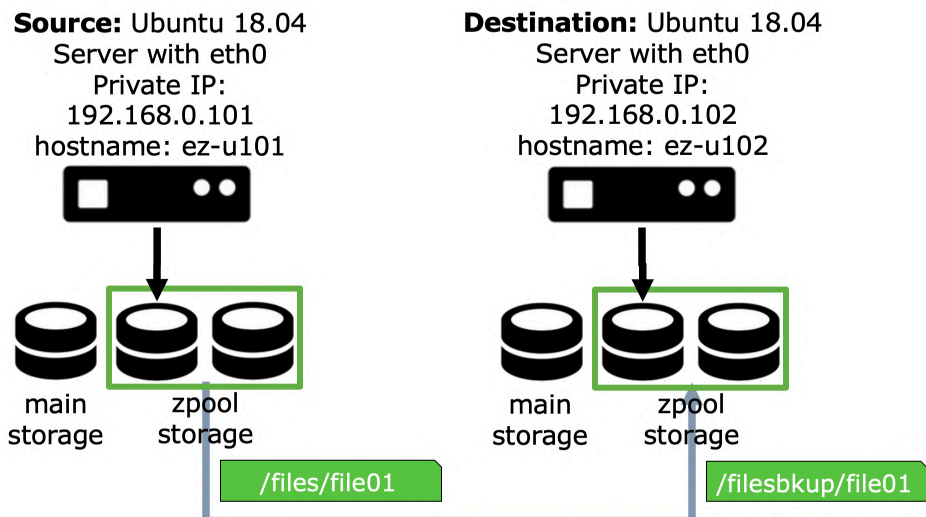
A Data Replication Scenario on Ezstackr



Will use ZFS to send from Source's /files/file01 to Destination's /filesbkup/file01



Sending file01 from Source to Destination



```
sudo zfs send -R files@snapshot2 | ssh ubuntu@192.168.0.102 sudo zfs receive -Fu filesbkup
```

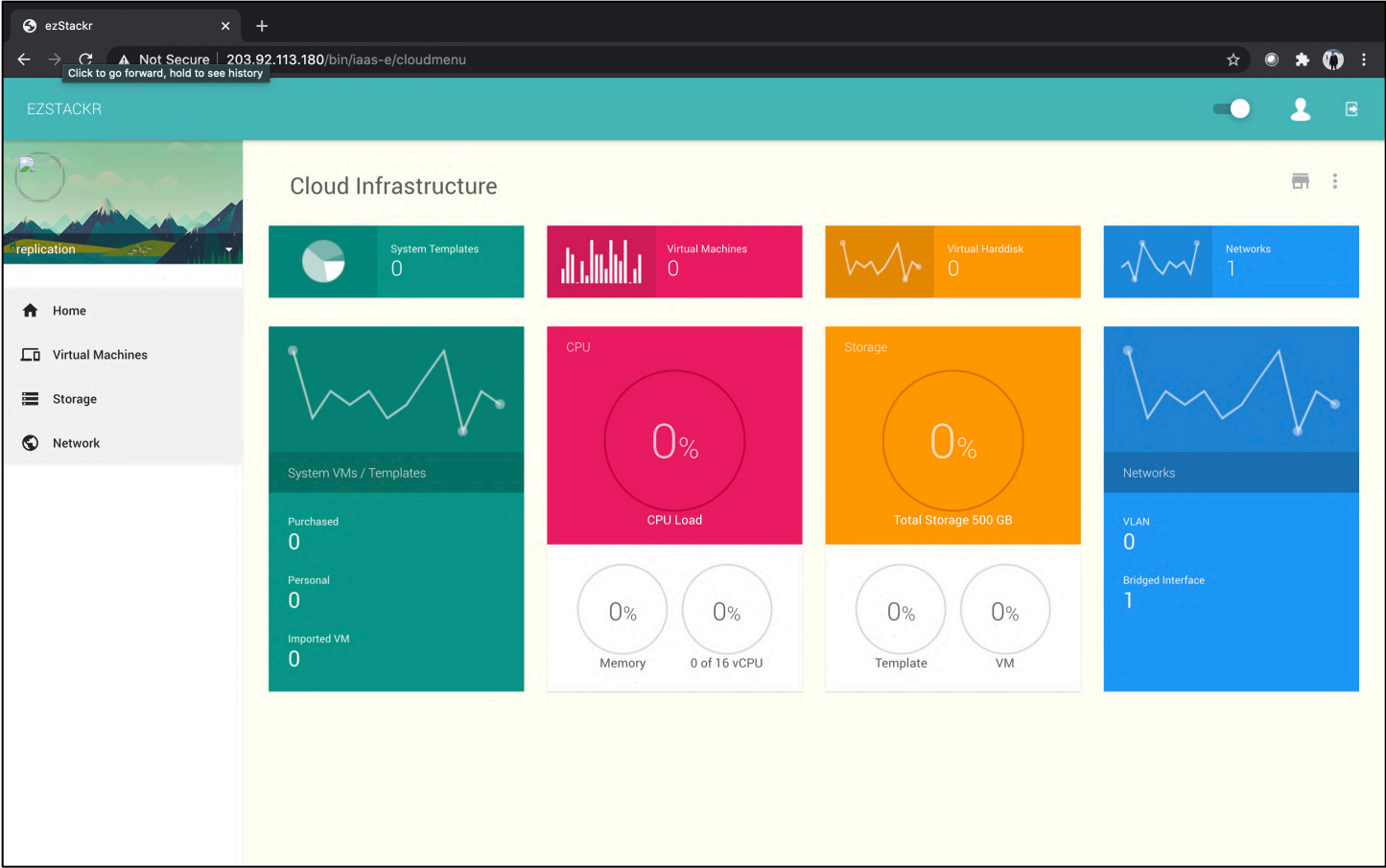
References:

Many Ezstackr operations have already been described in the technical guide #1. Please refer to guide #1 for details.

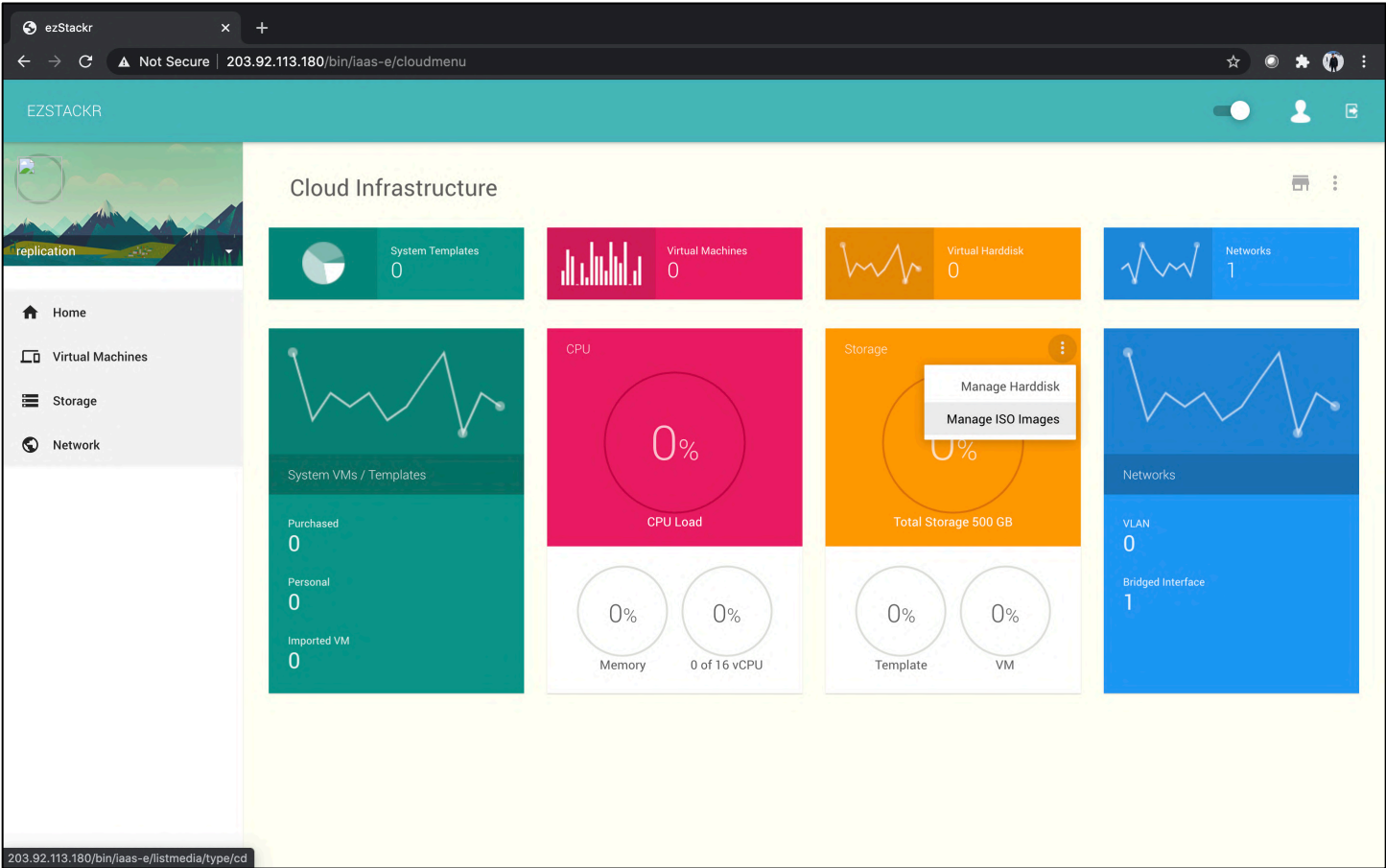
CREATING TWO NEW VMs (SOURCE AND DESTINATION). INSTALLING UBUNTU 18.04 SERVER ONTO BOTH VMs

LOGGING IN:

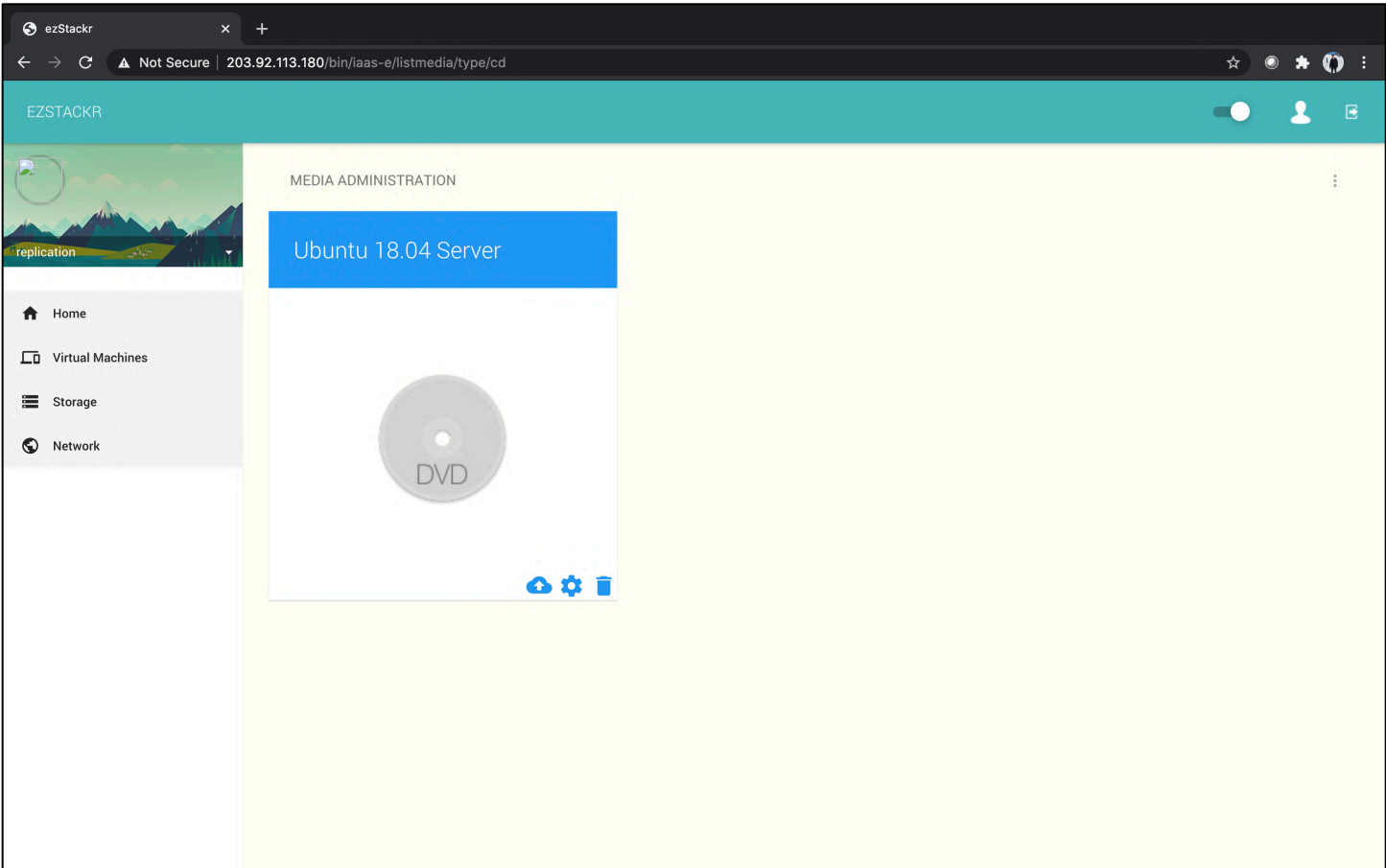
As User: Log in



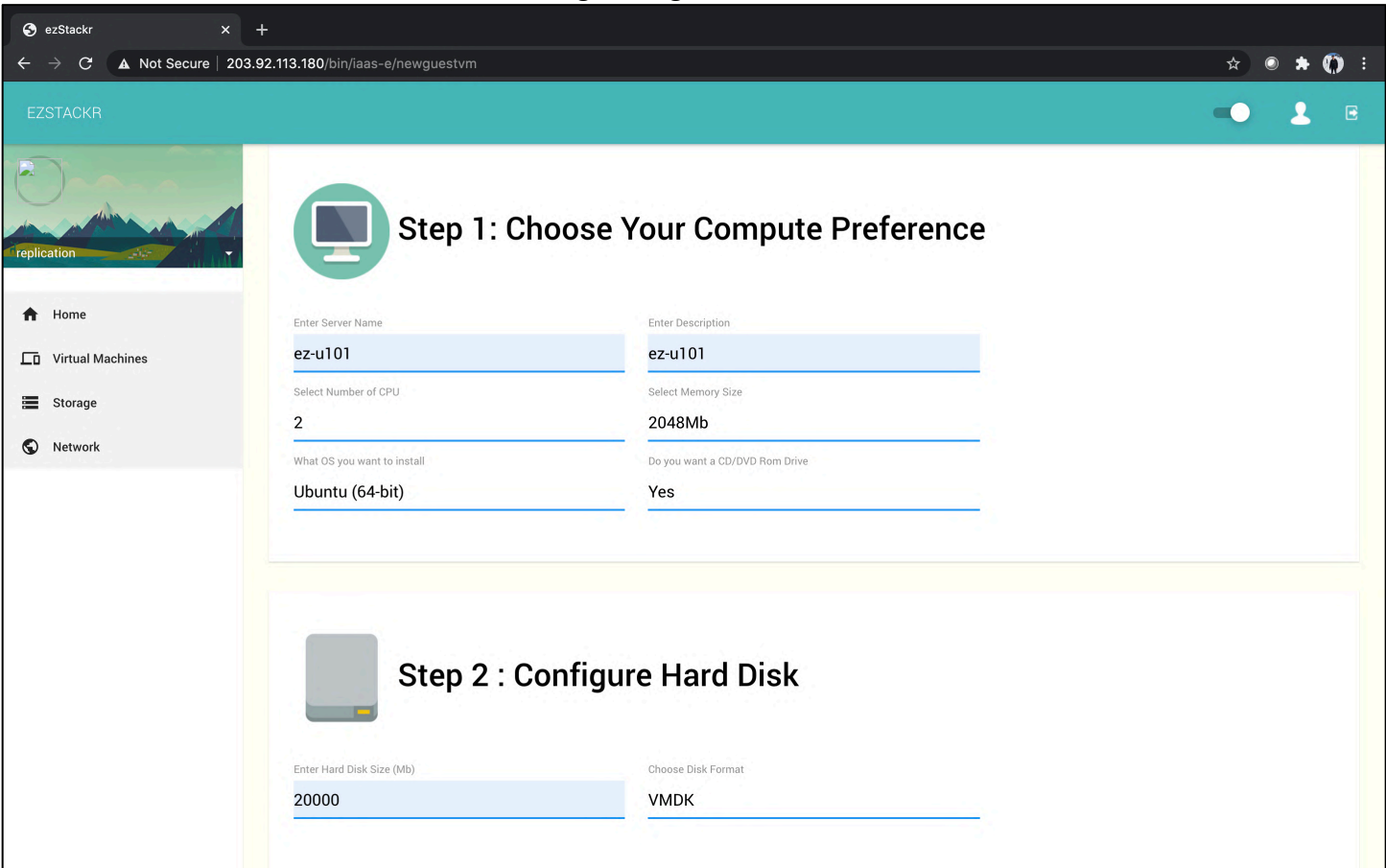
As User: Select “Manage ISO Images” and upload Ubuntu 18.04 ISO disks or lookup the ISO disks available in your account:

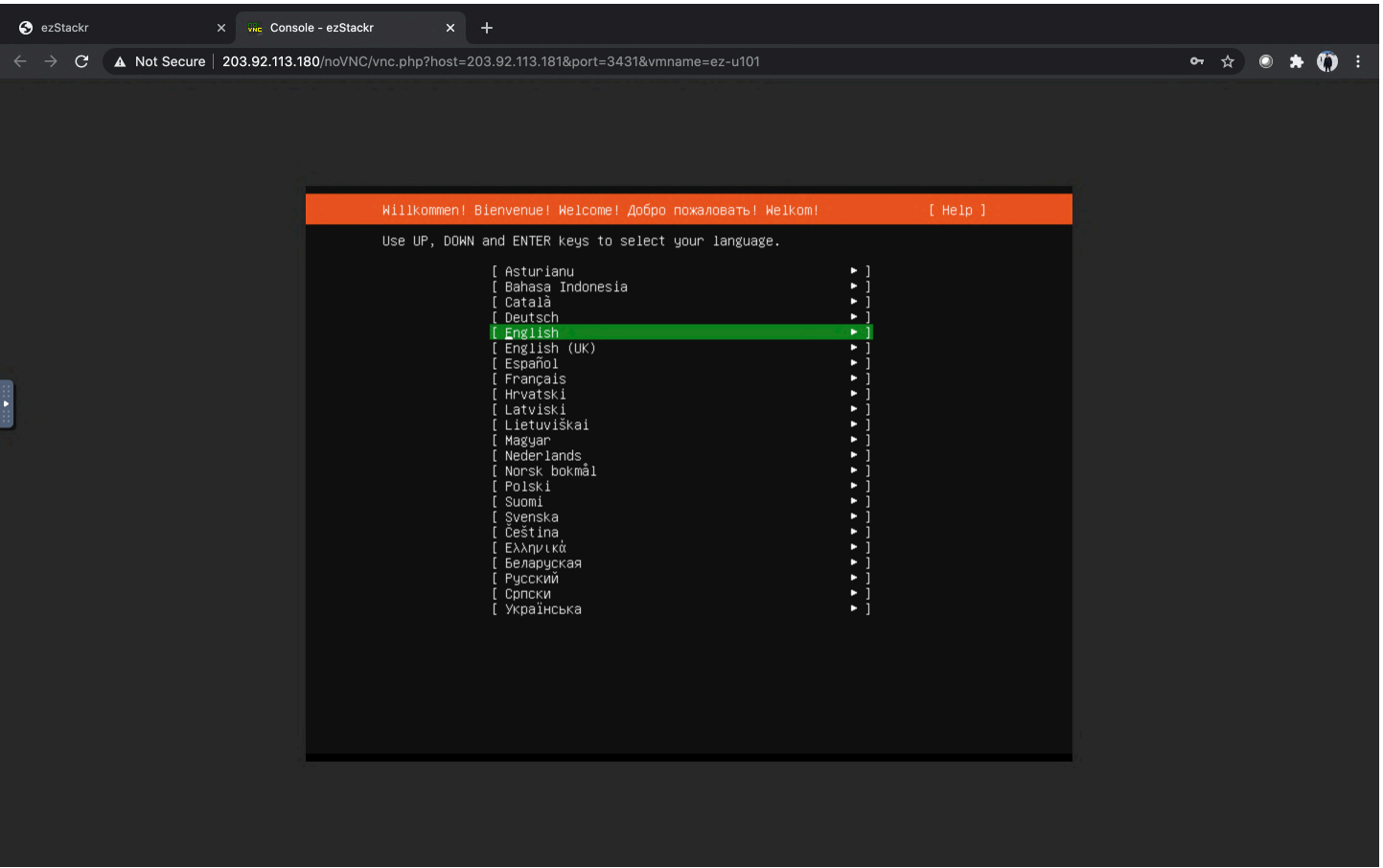
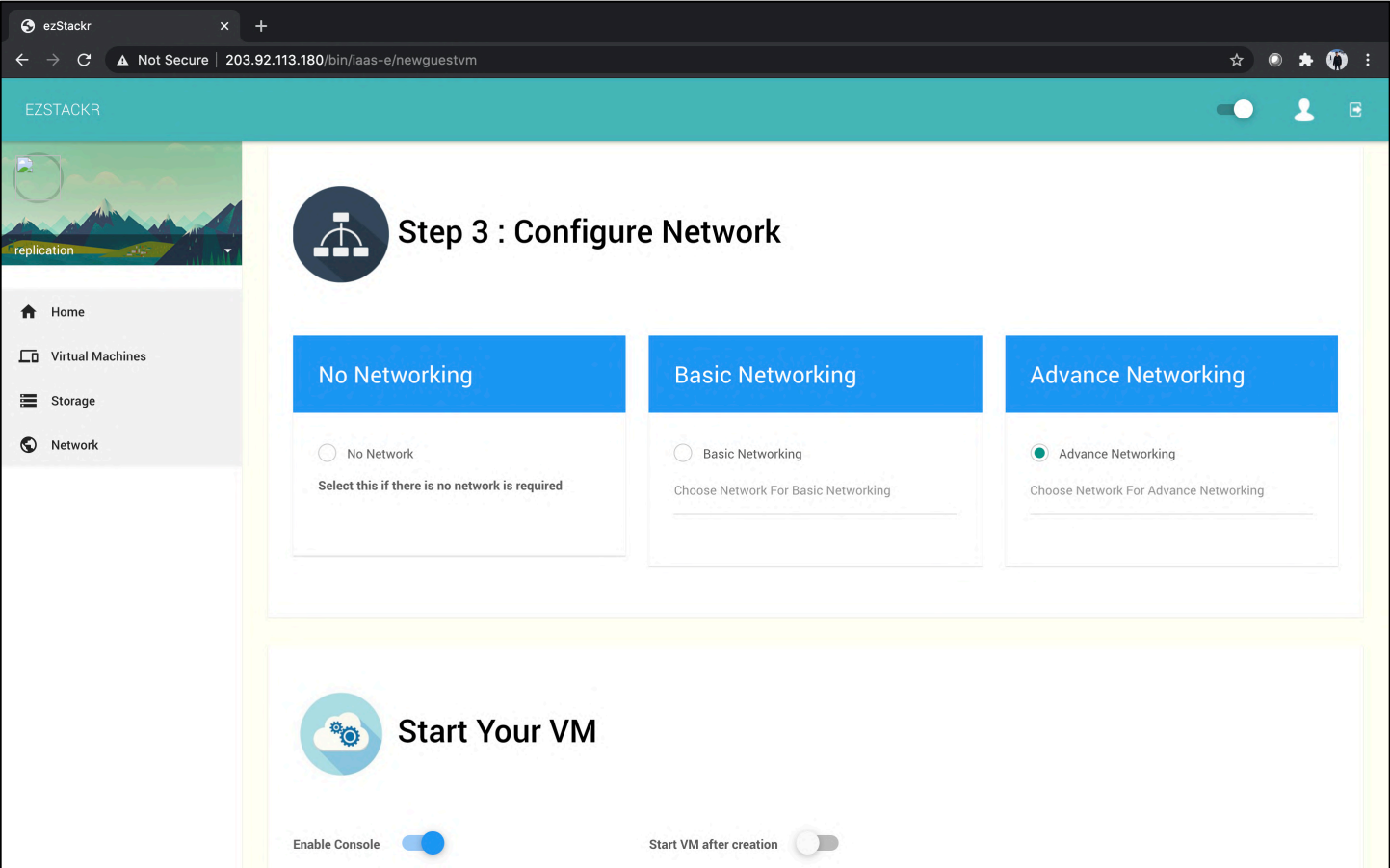


As User: View available ISO disks previously uploaded

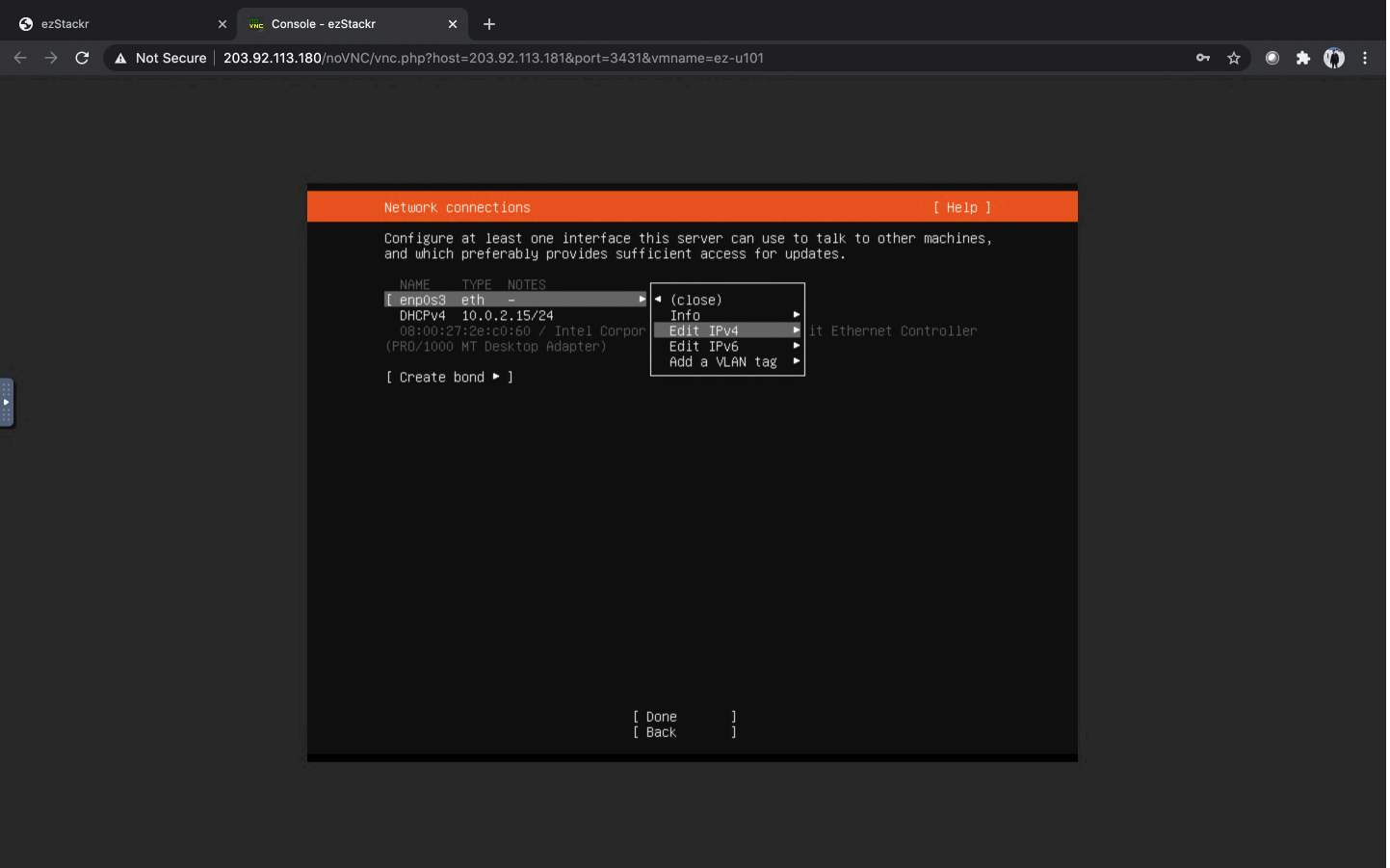


As User: Select “Virtual Machines” and “Launch New VM”. Install VM (this is the Source VM) with Ubuntu 18.04 DVD-ROM distribution with the following settings:

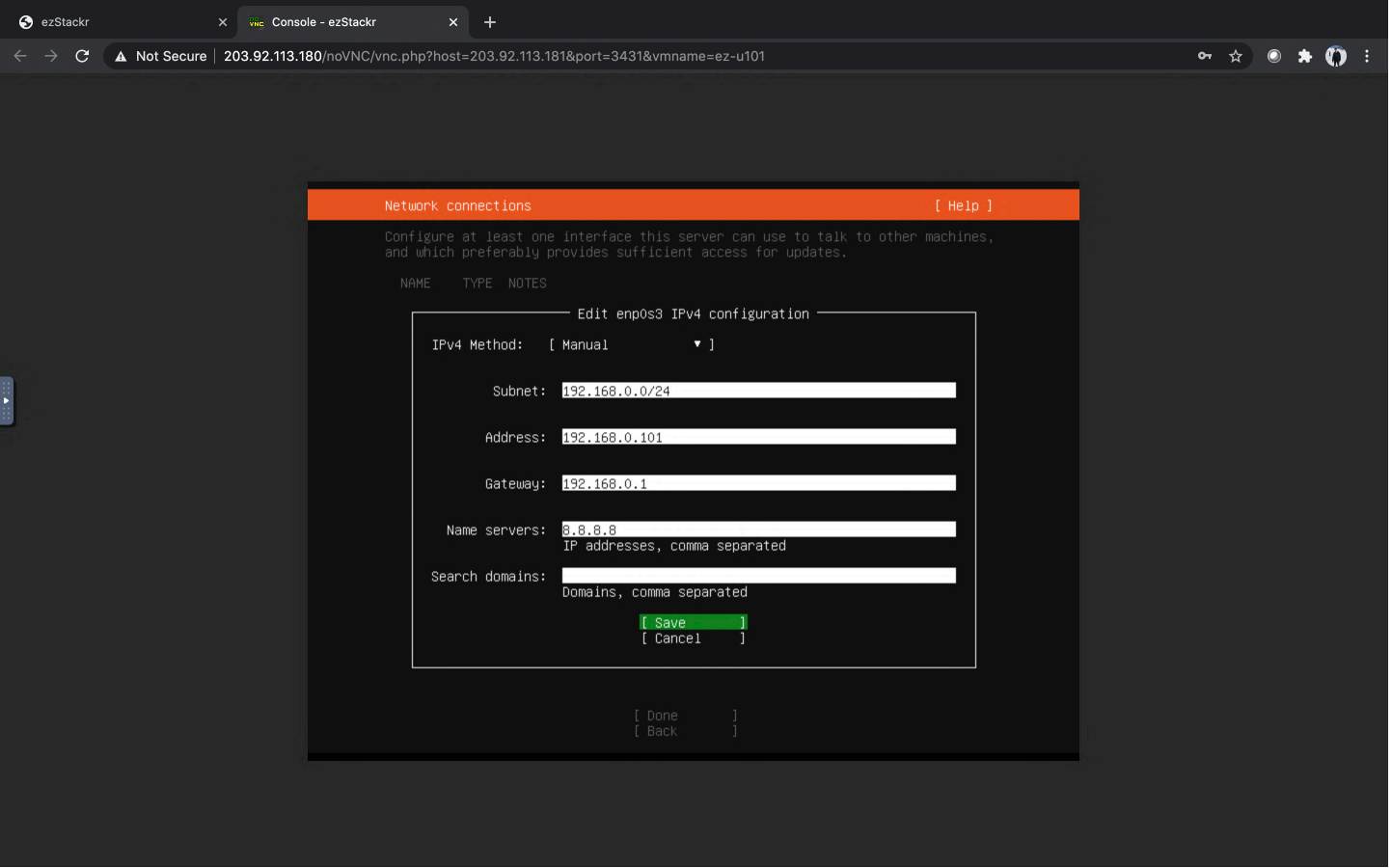




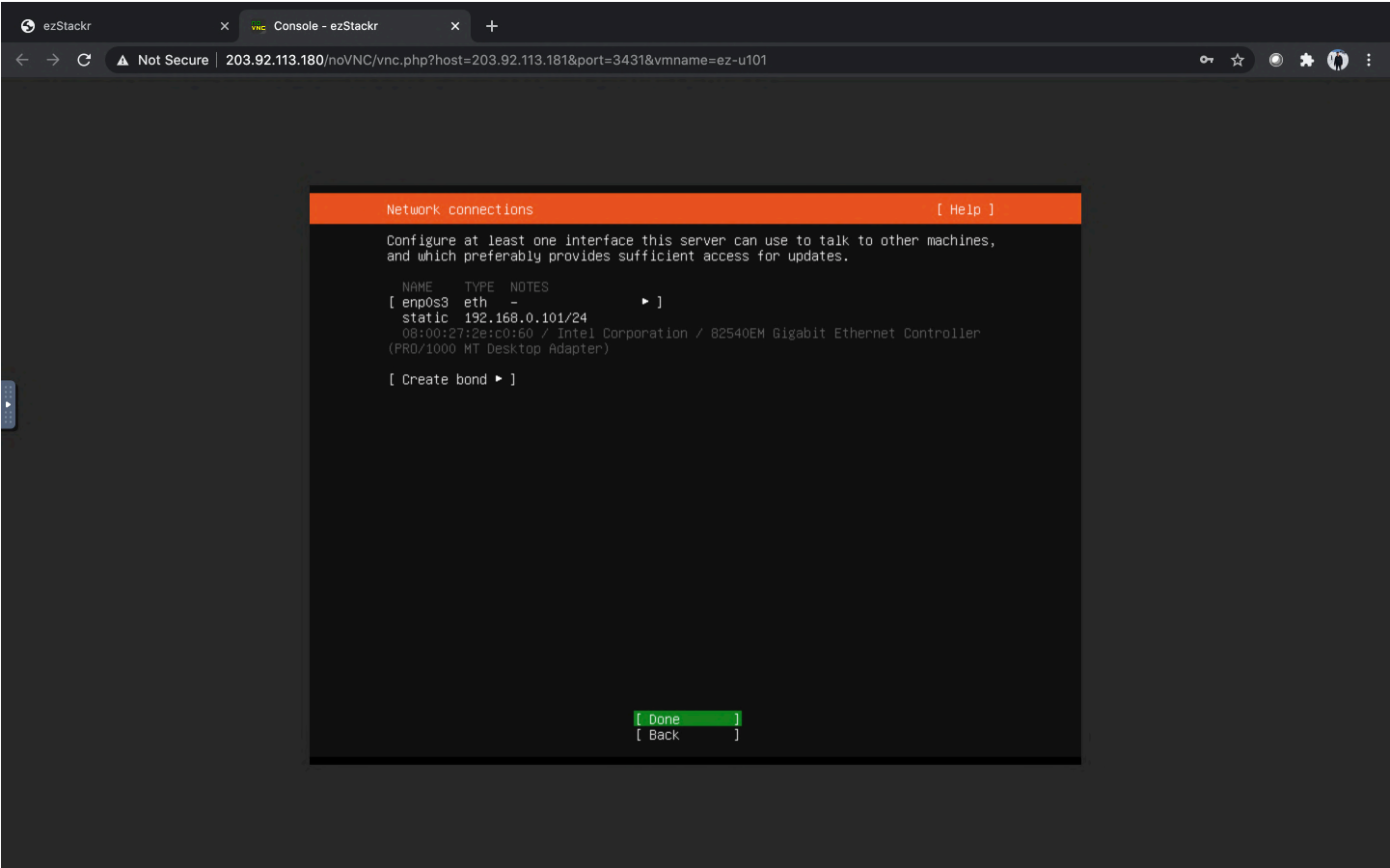
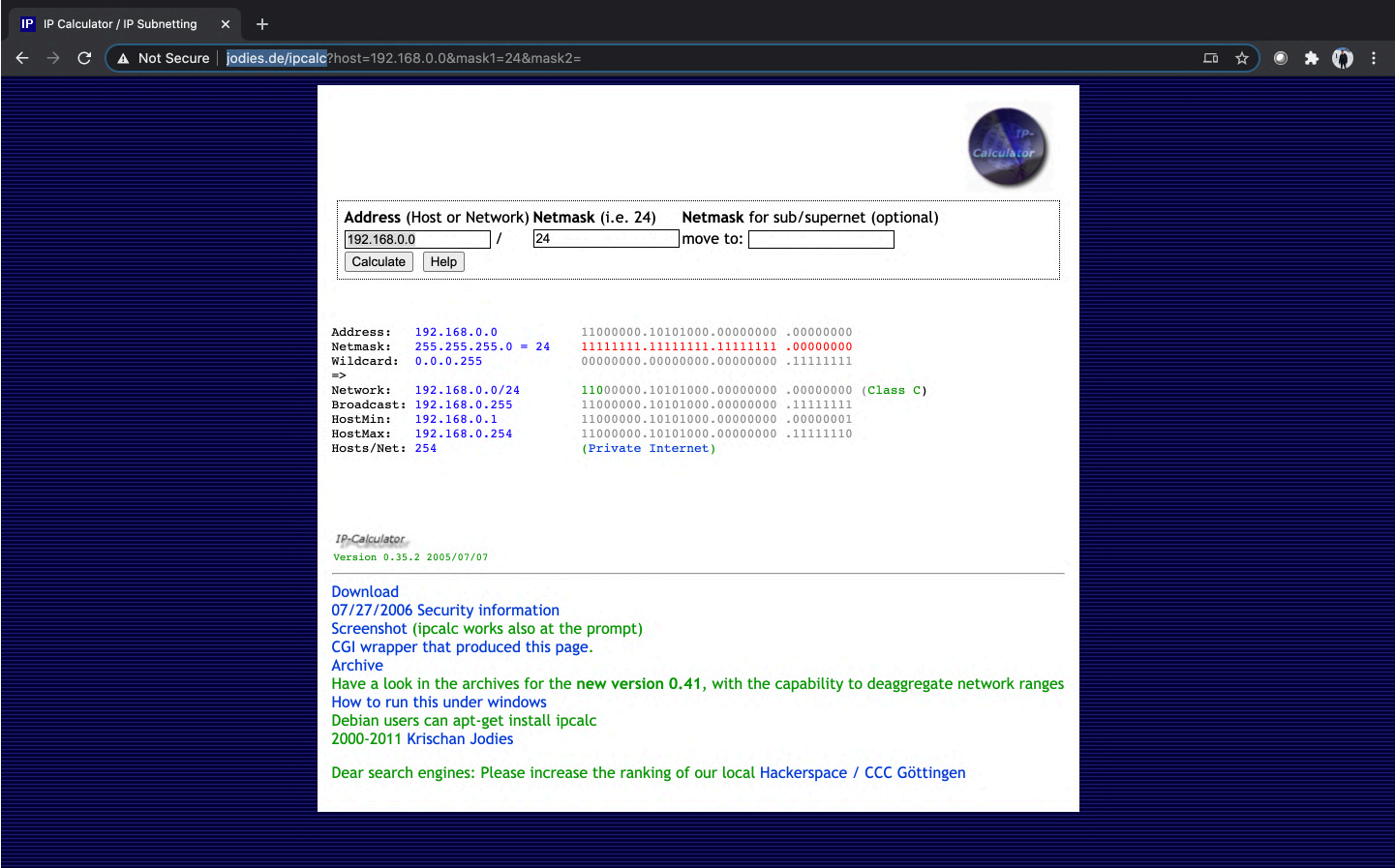
As User: Edit the IP settings during install



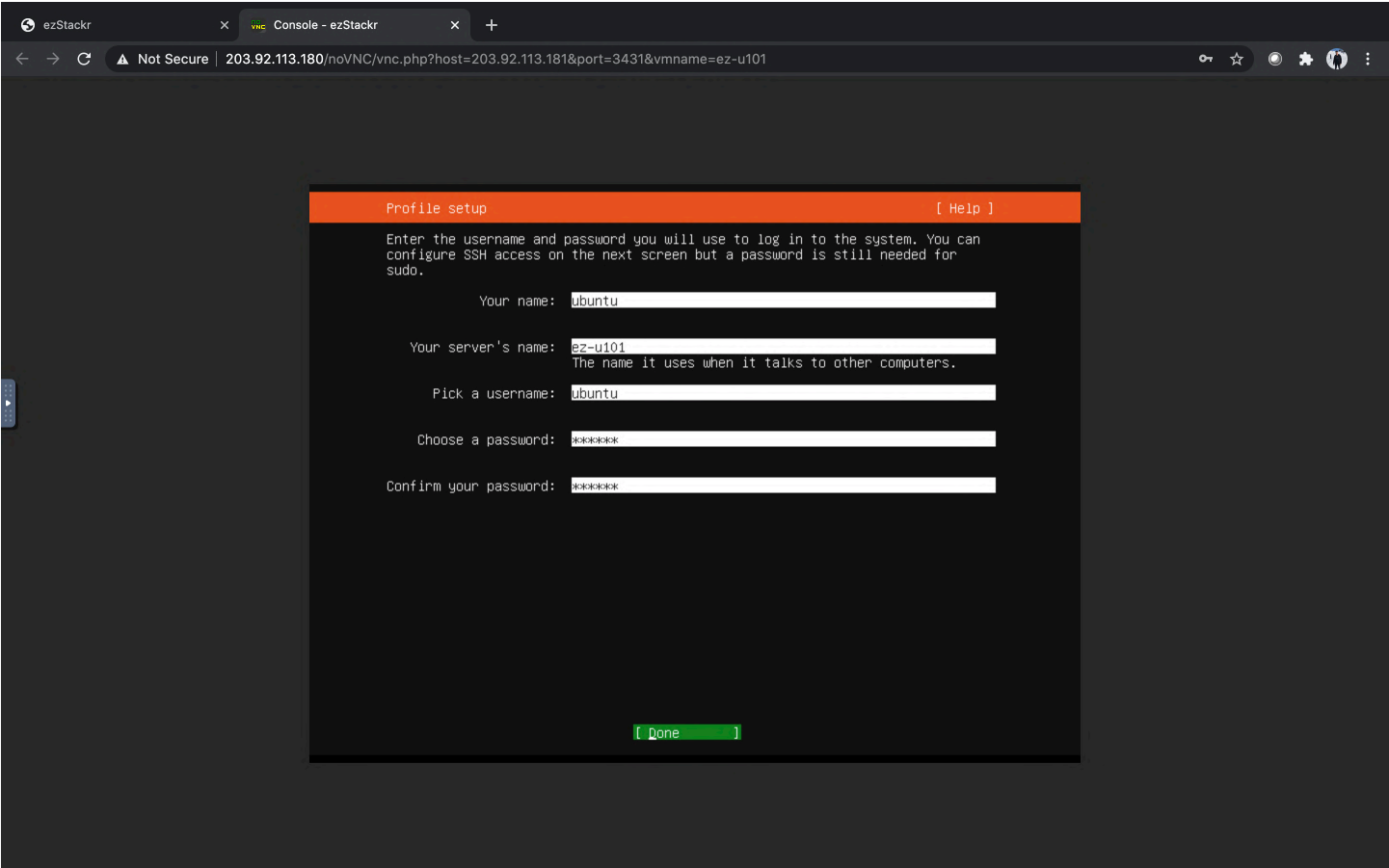
As User: Enter IP settings for Ubuntu network



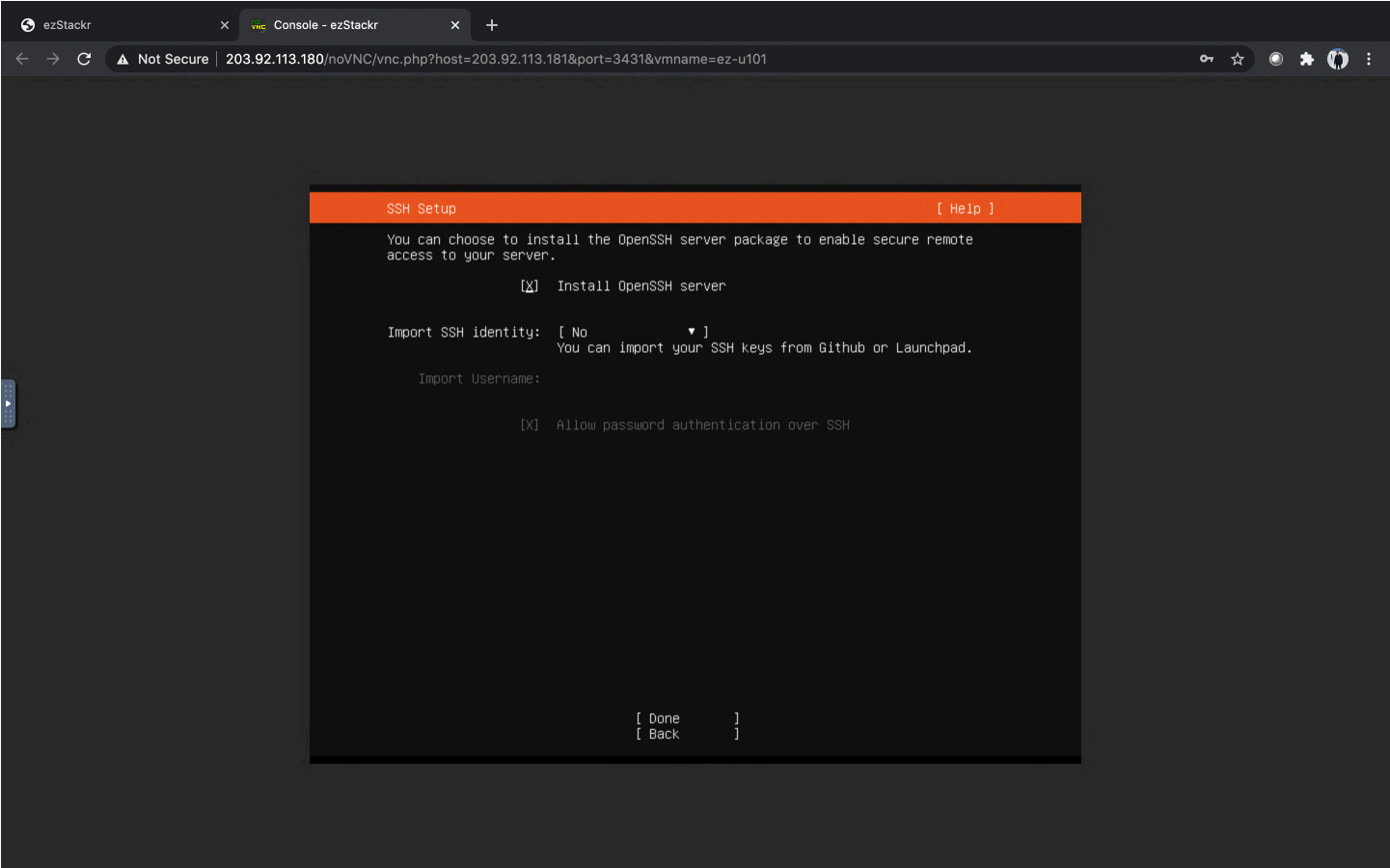
The above IP can be calculated using online tools such as: <http://jodies.de/ipcalc>



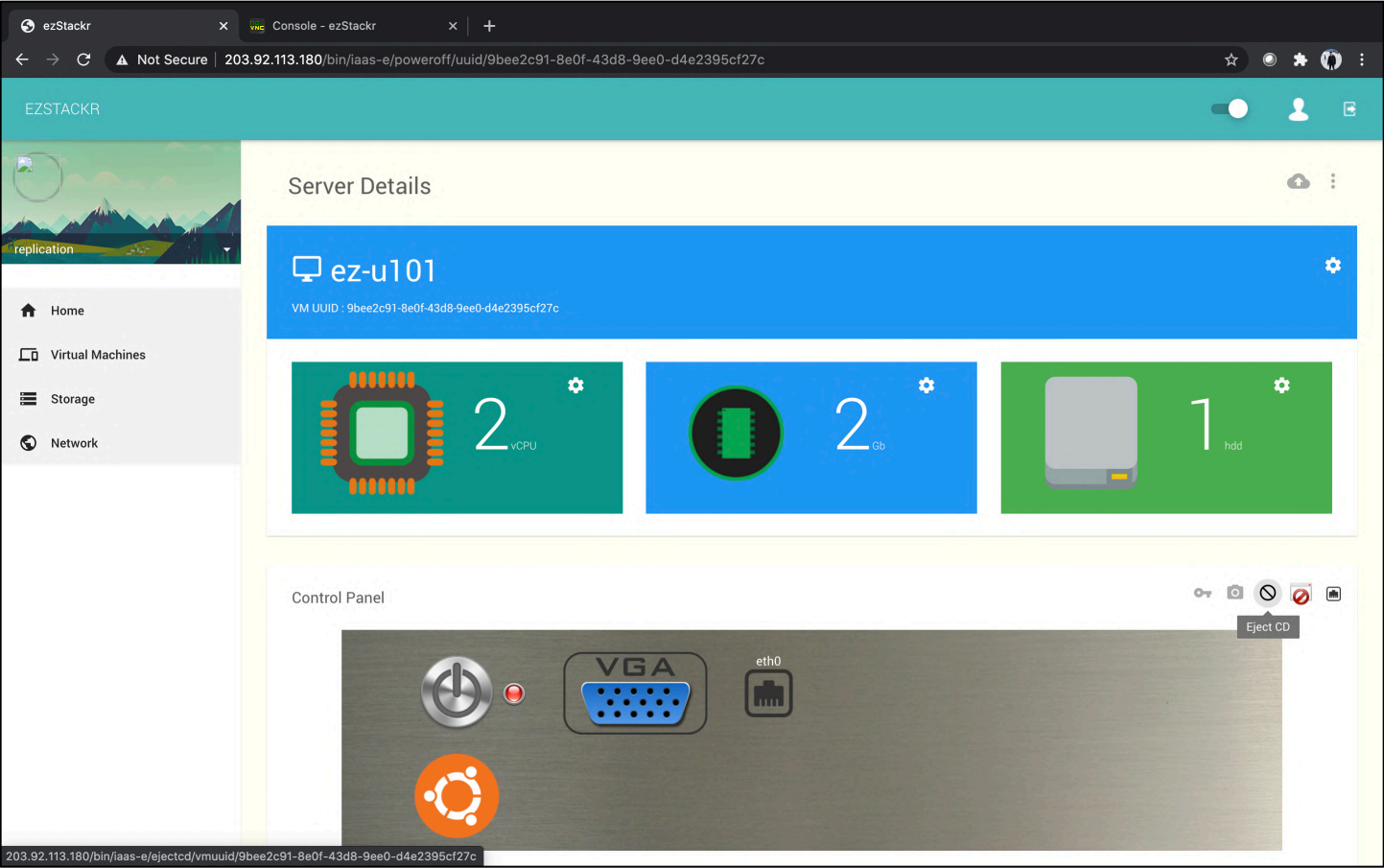
As User: Enter settings for the Ubuntu user:



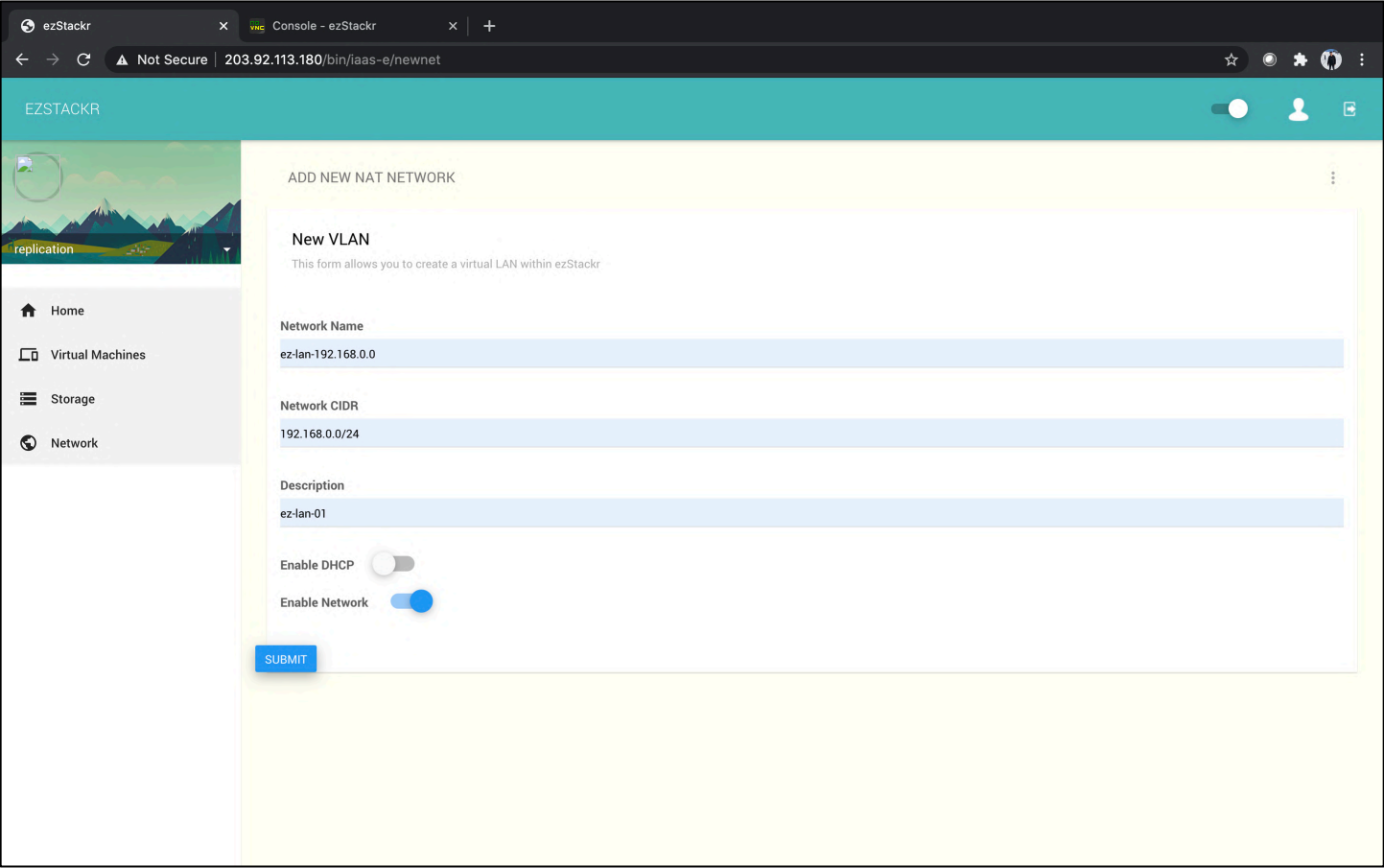
As User: Select to install OpenSSH server



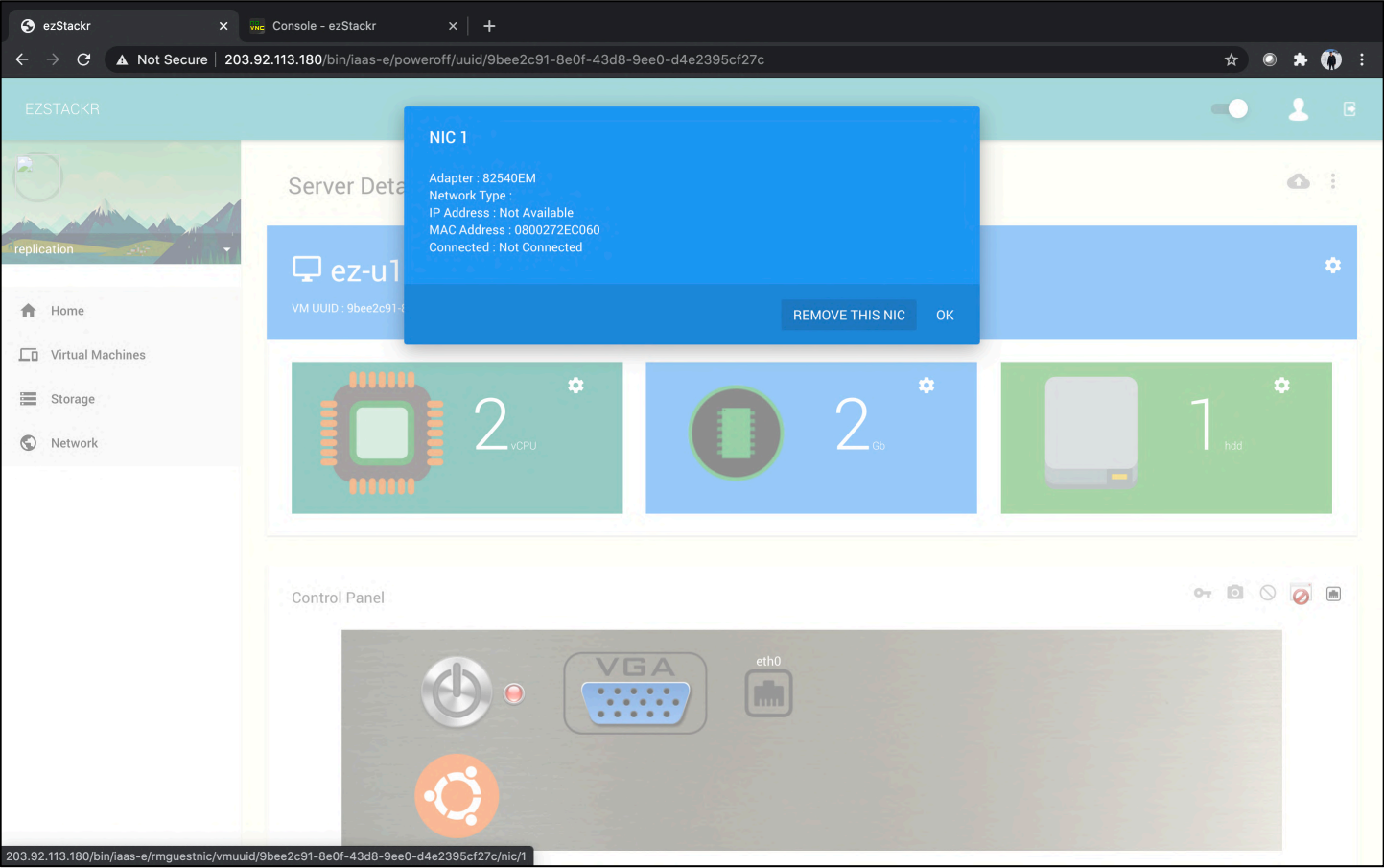
As User: When installation completes, shutdown and eject the DVD-ROM



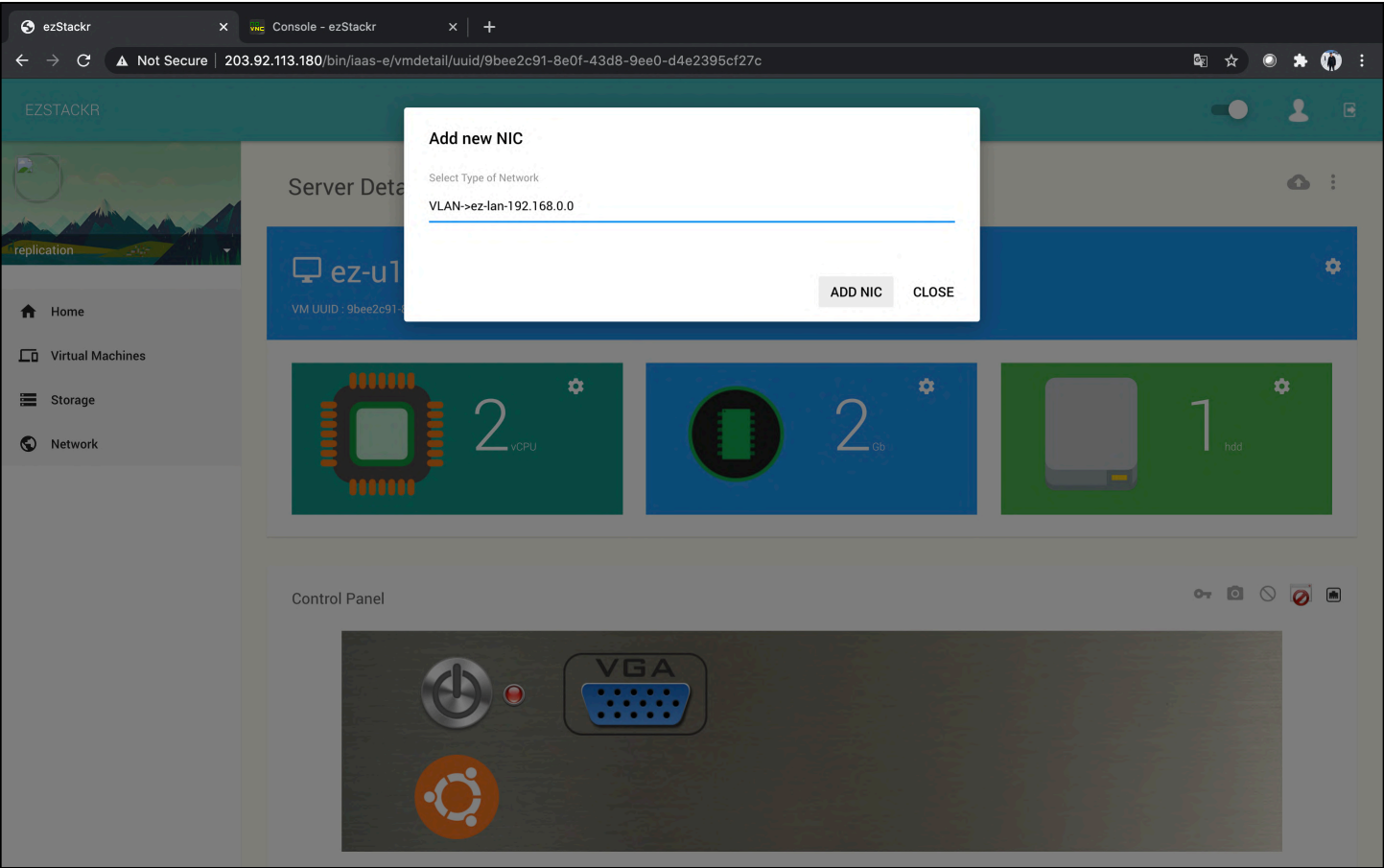
As User: Select “Network” and add new VLAN / NAT network



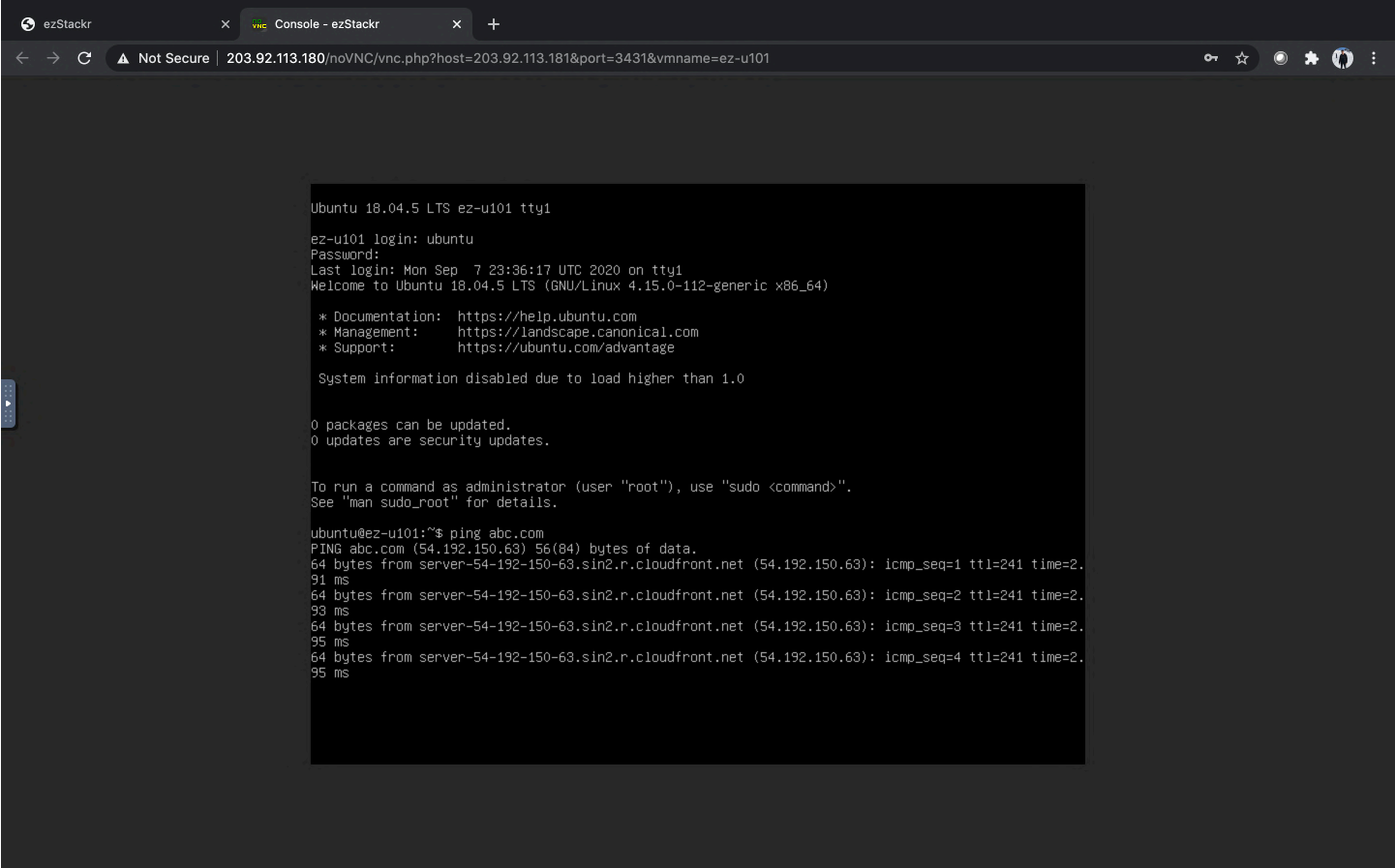
As User: Remove the default NIC



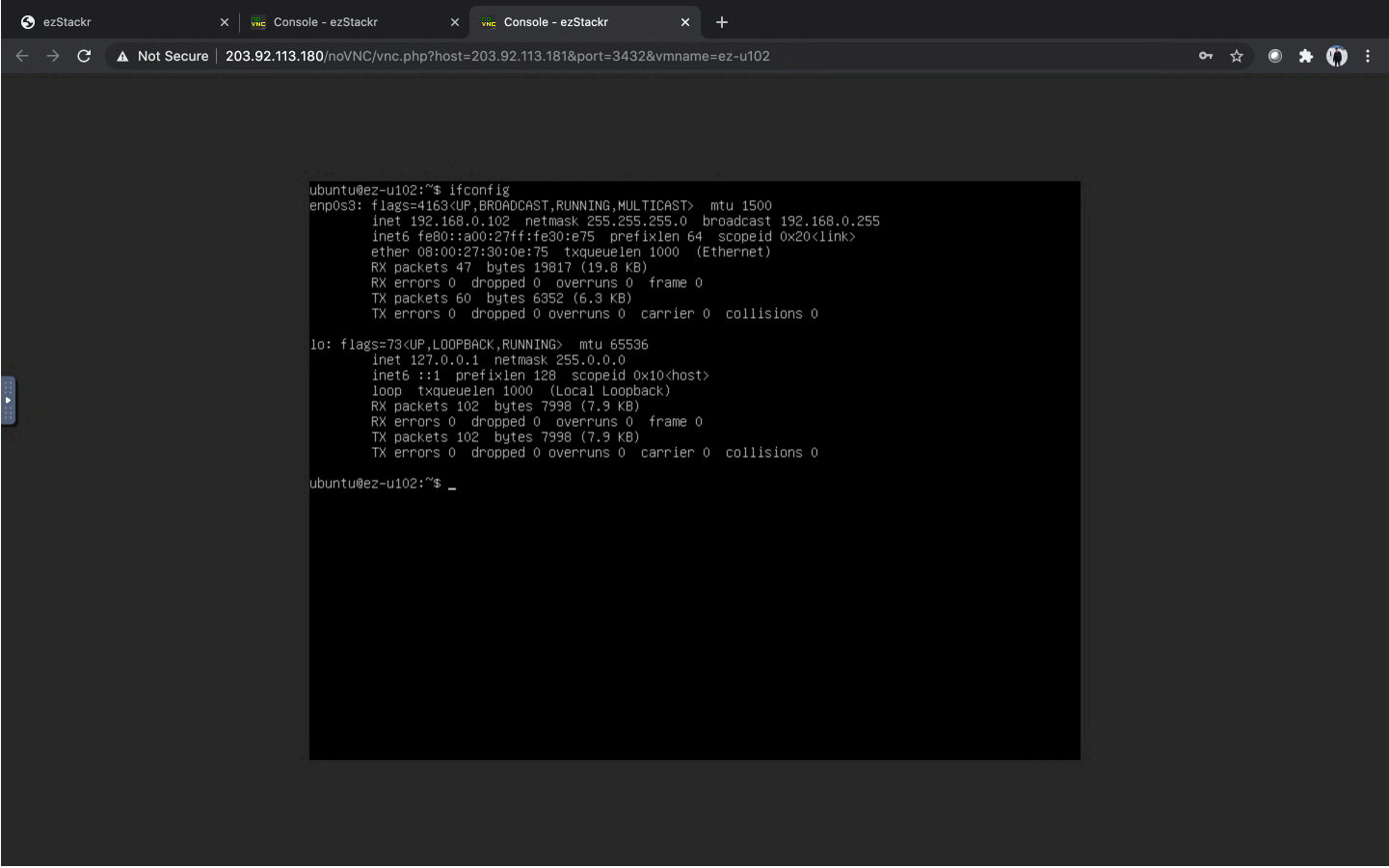
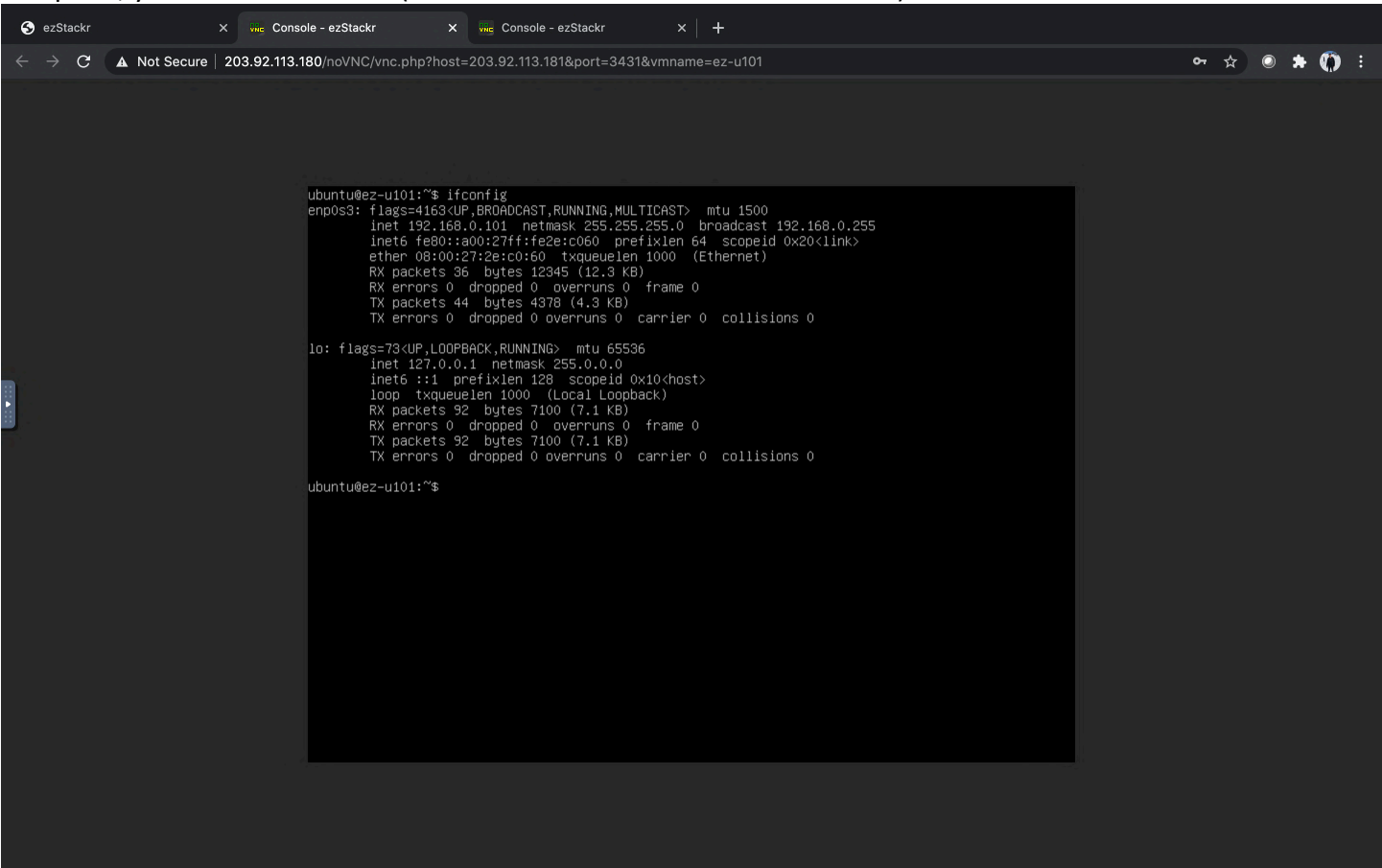
As User: Add newly created VLAN / NAT network



As User: Start up the VM and test the network



As User: Create and install the second VM (the Destination VM) using the same steps above. When this is complete, you now have 2 VMs (the Source VM and the Destination VM)



As User: Ensure that the appropriate set up is correctly applied to both VMs e.g.

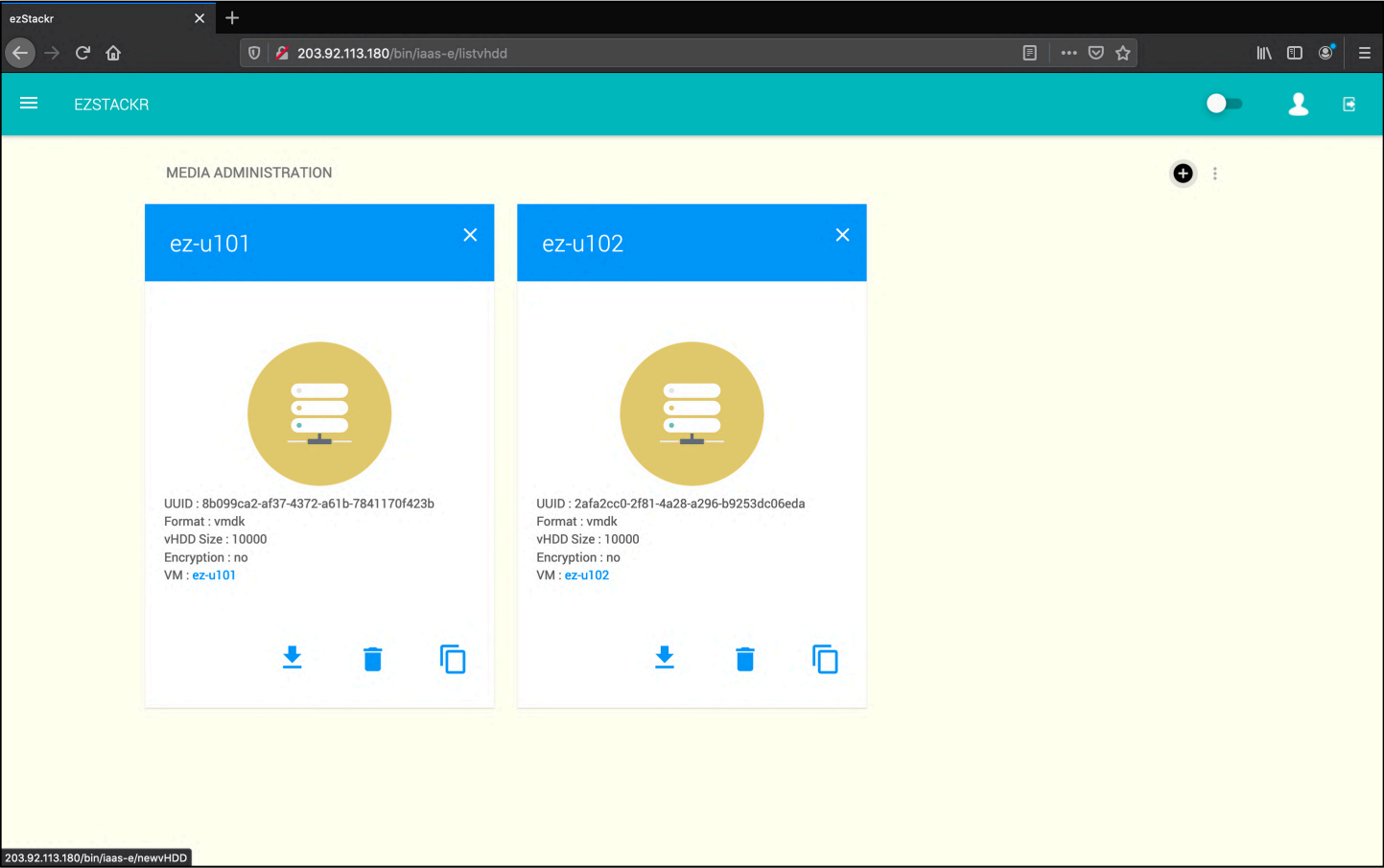
Change Terminal Font:
sudo dpkg-reconfigure console-setup
change to TerminusBold 12x24

Configure time:
timedatectl set--ntp yes
timedatectl set--timezone Asia/Singapore

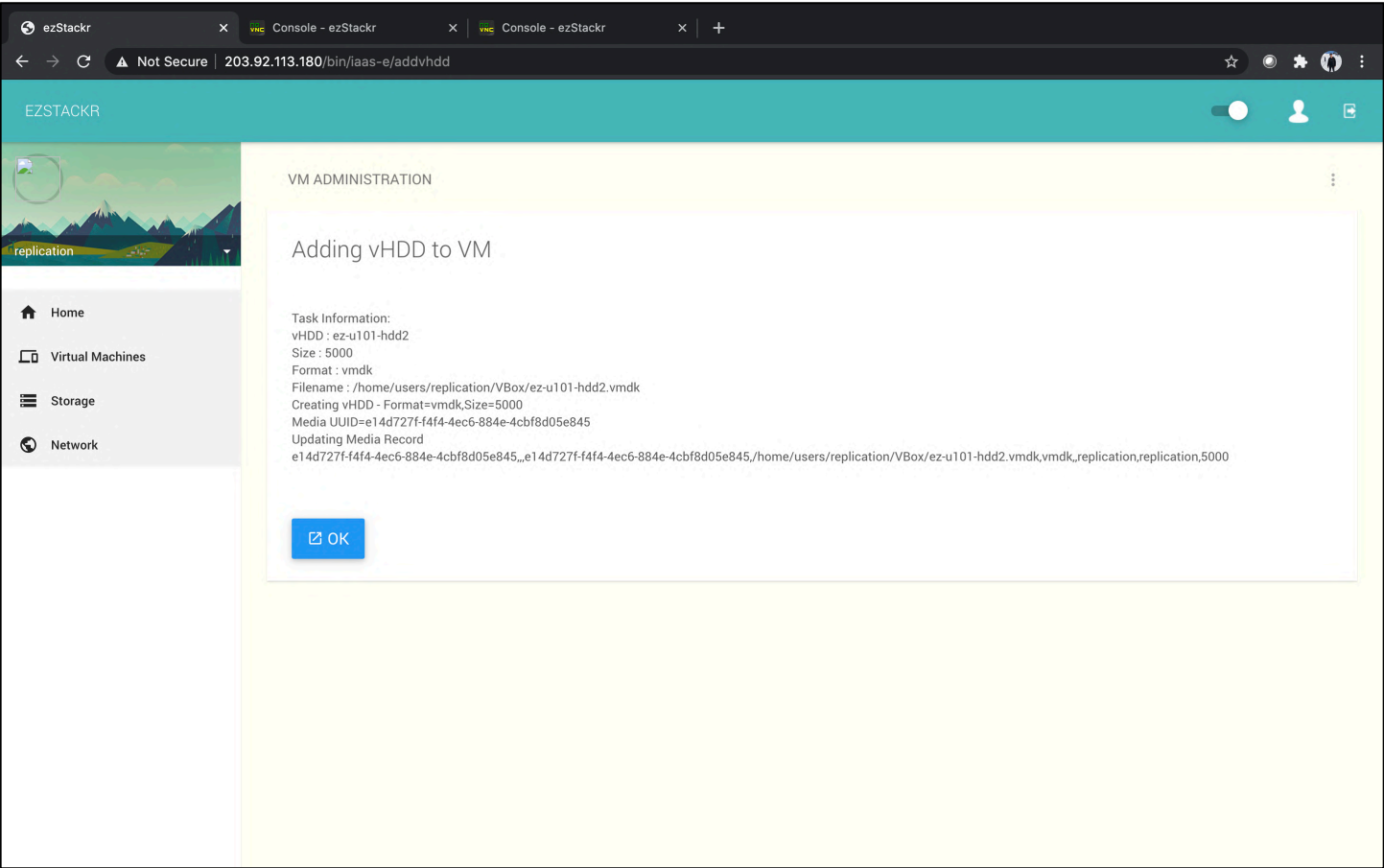
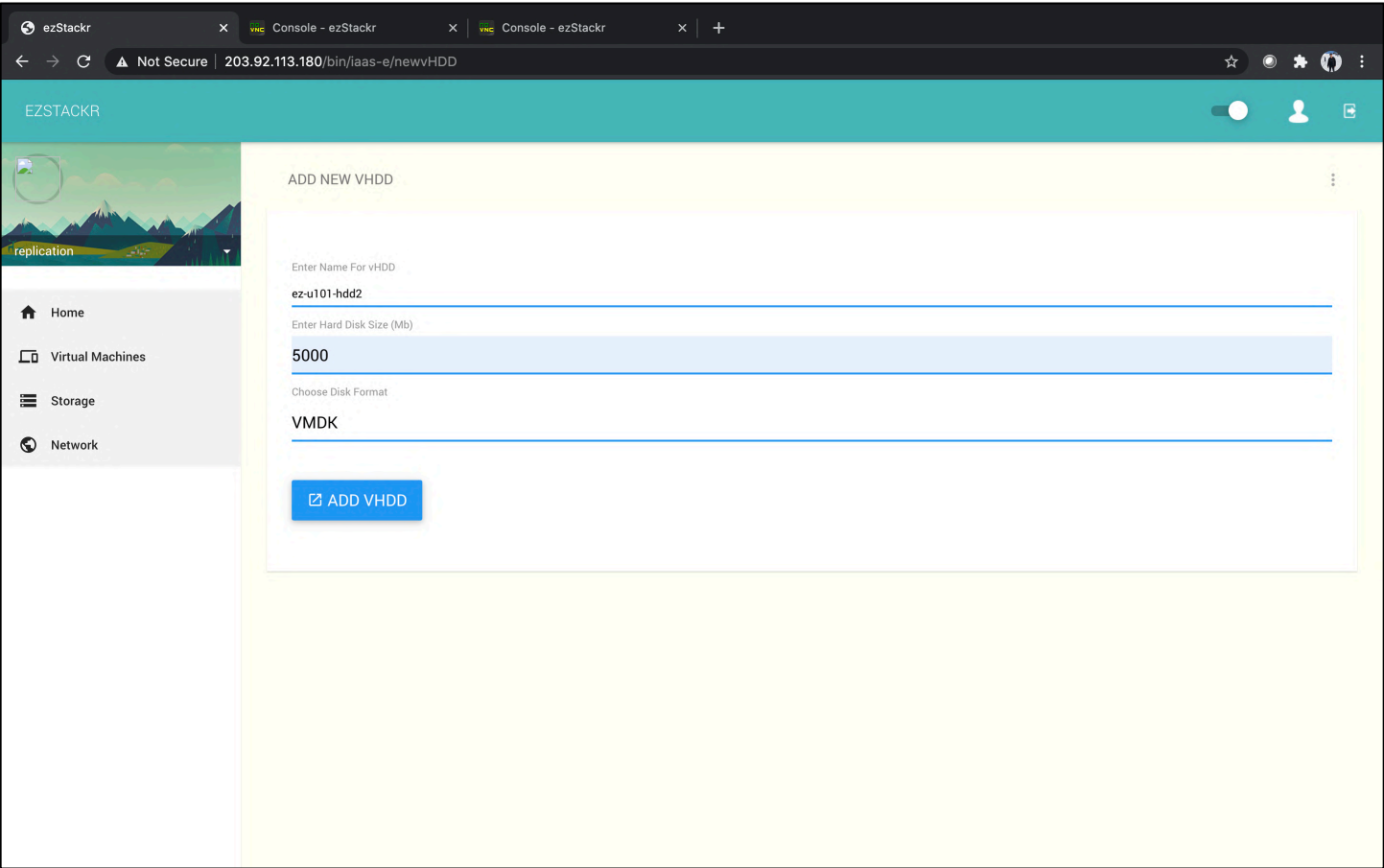
After configuration, shutdown both VMs

ADDING AND CONFIGURING STORAGE FOR SOURCE VM AND DESTINATION VM

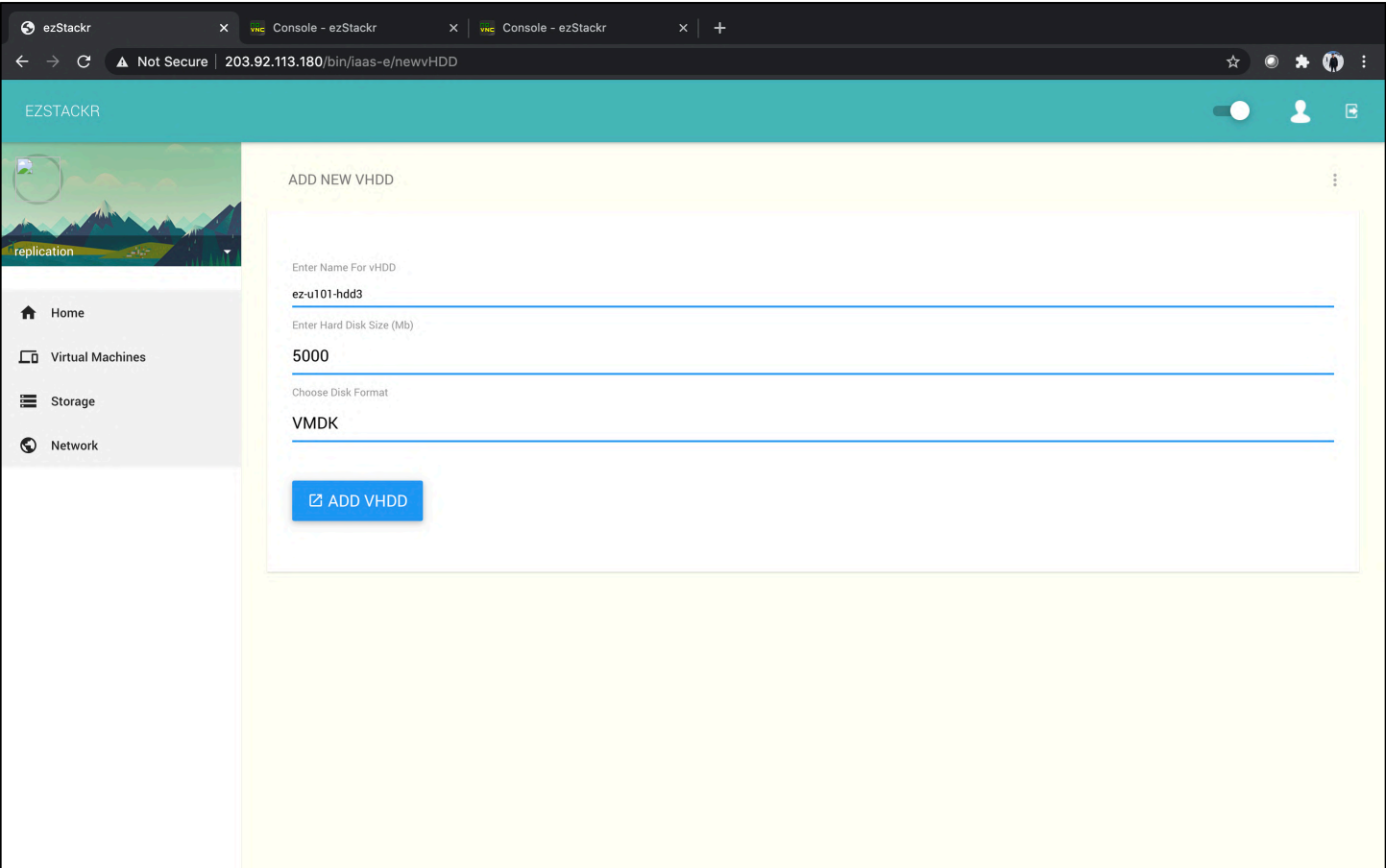
As User: Select “+” to add storage

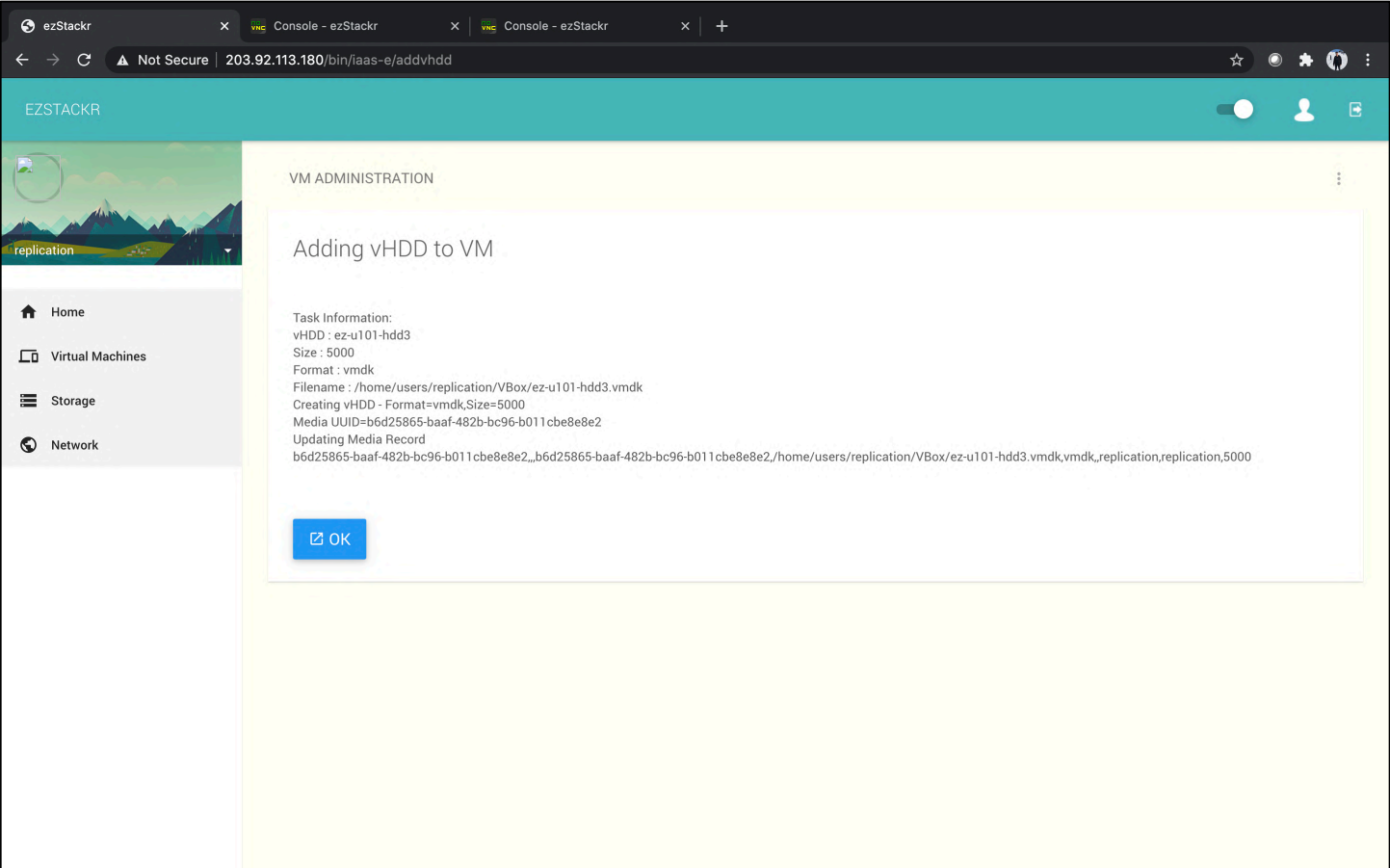


As User: Specify storage size for source and select “Add”

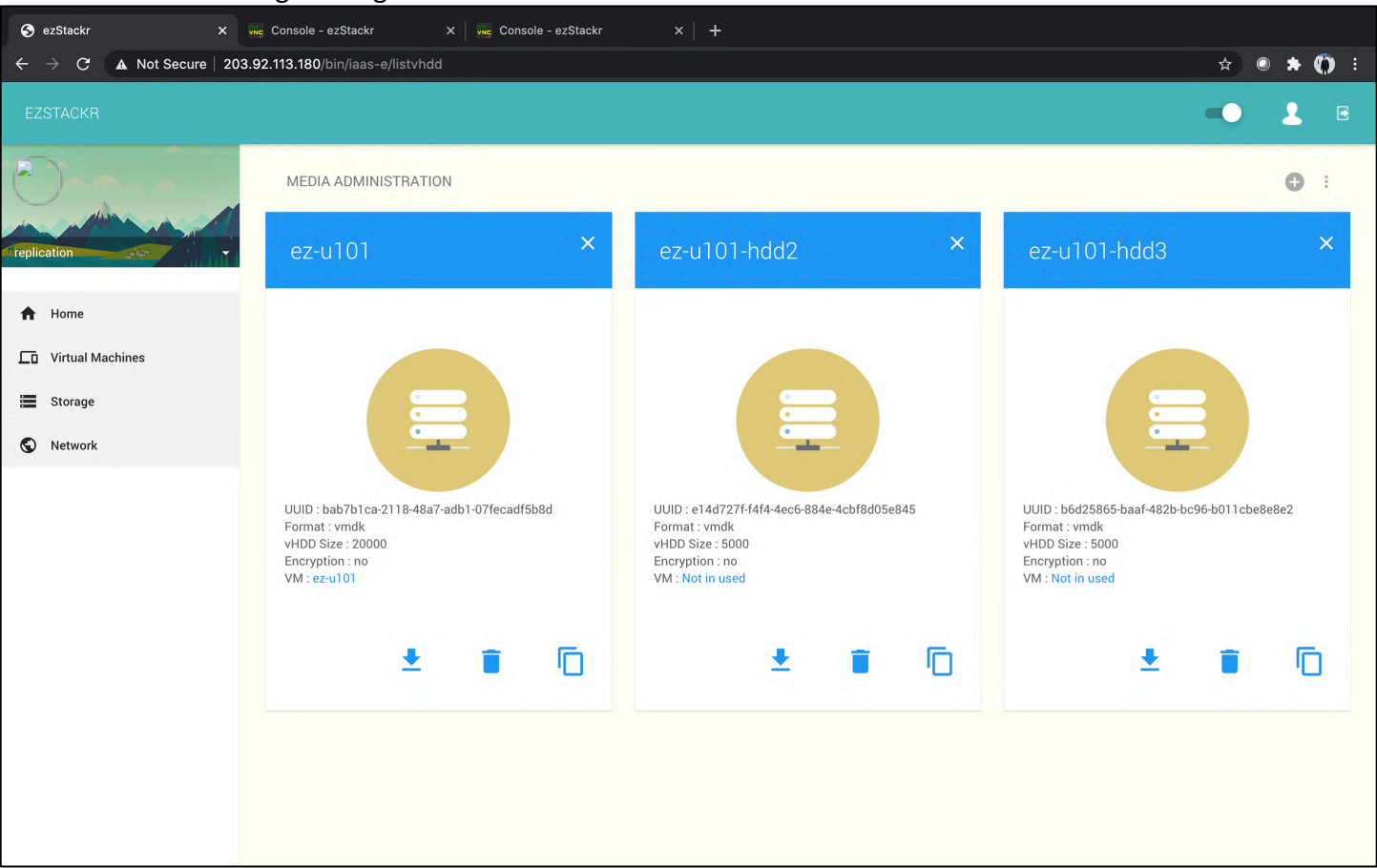


As User: Select “+” to add storage

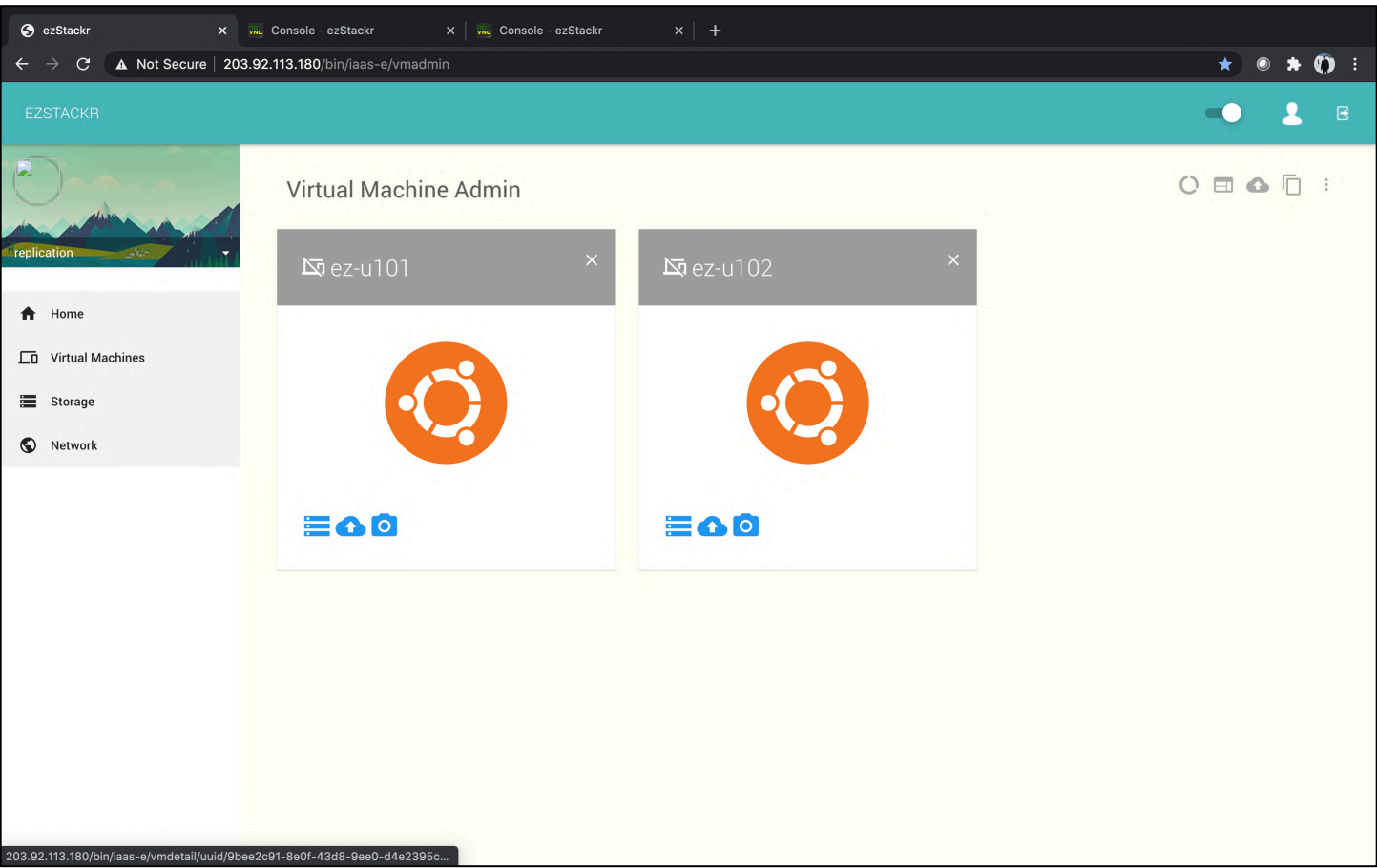




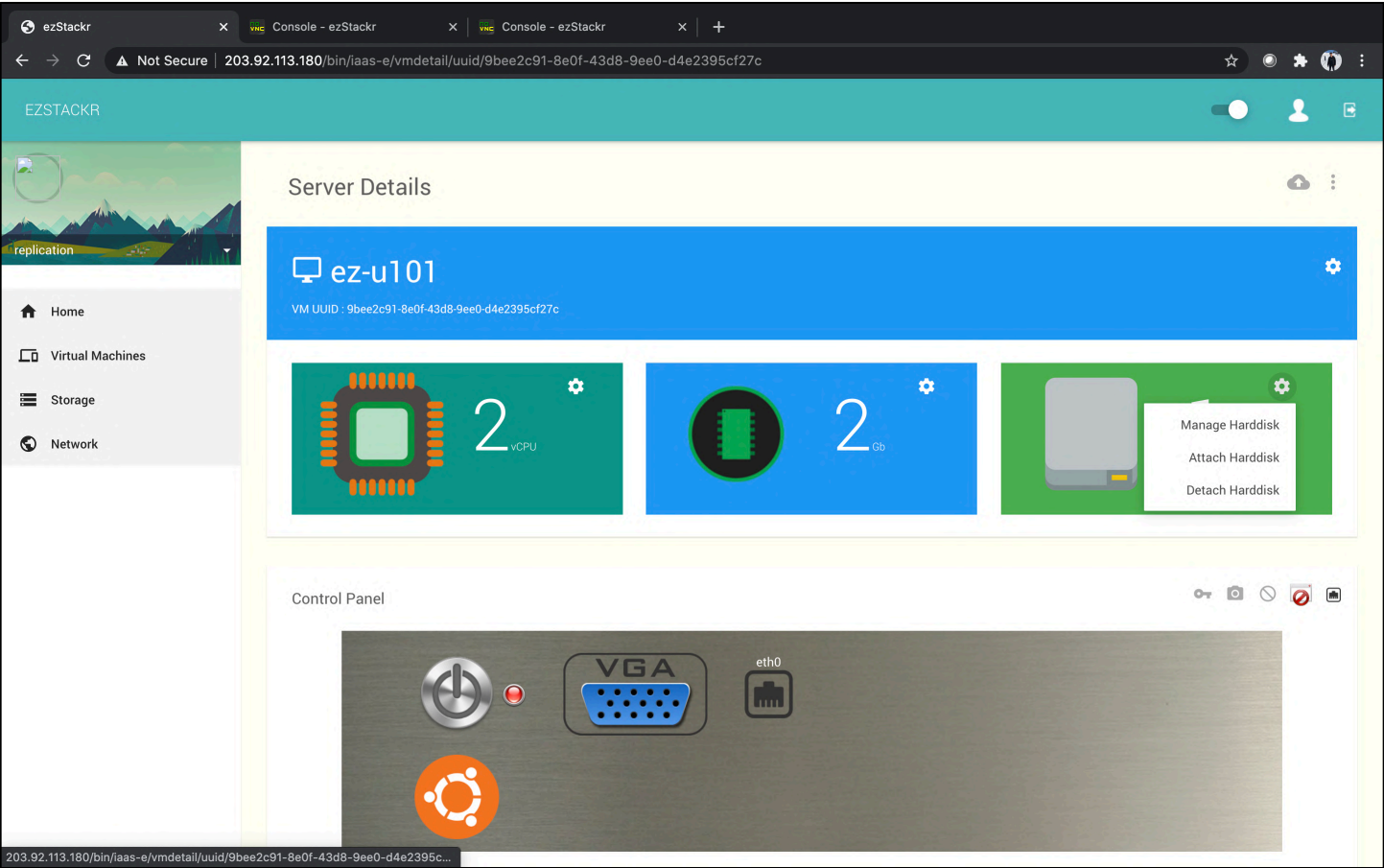
As User: View all storage configured



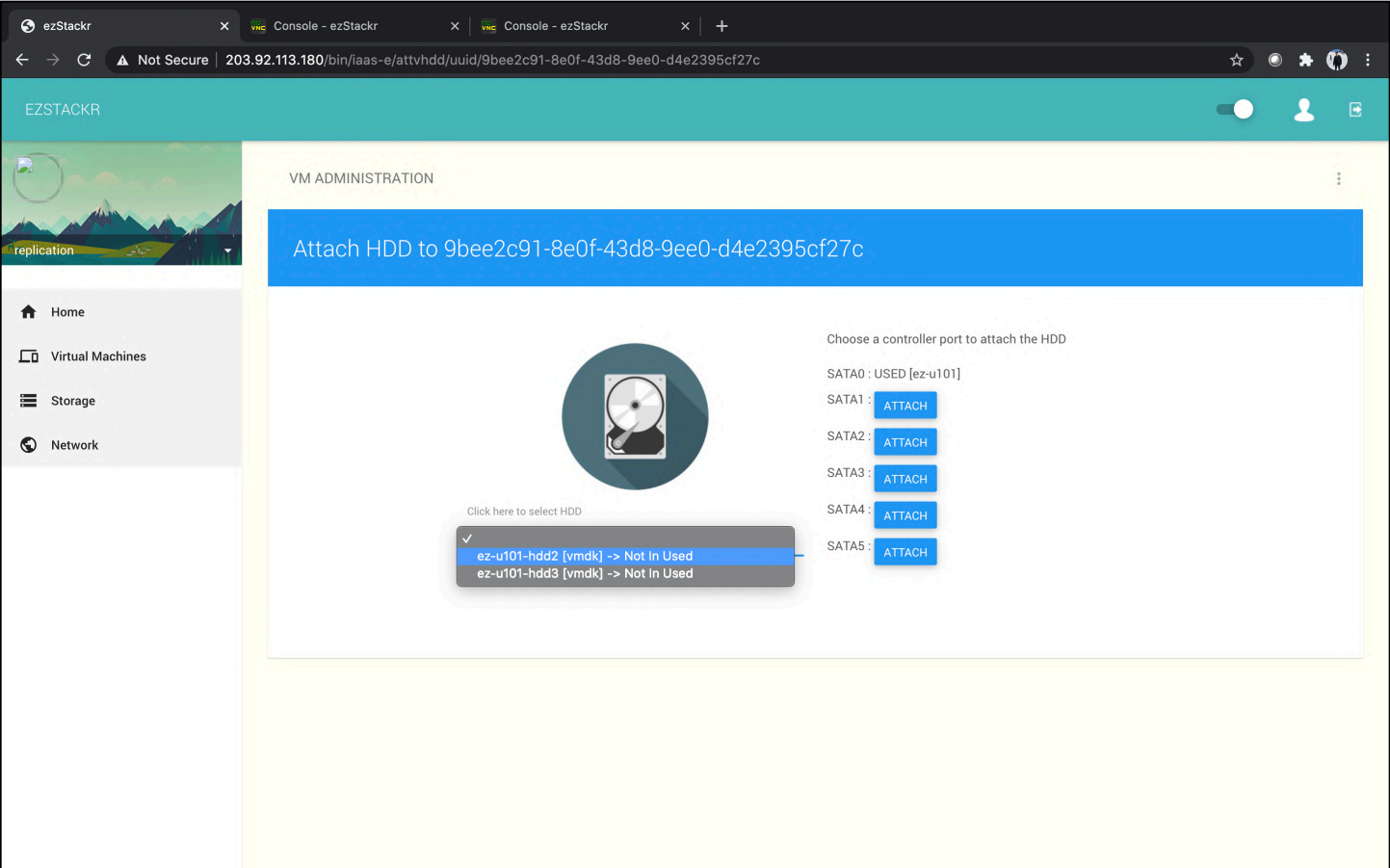
As User: Select “Virtual Machines”



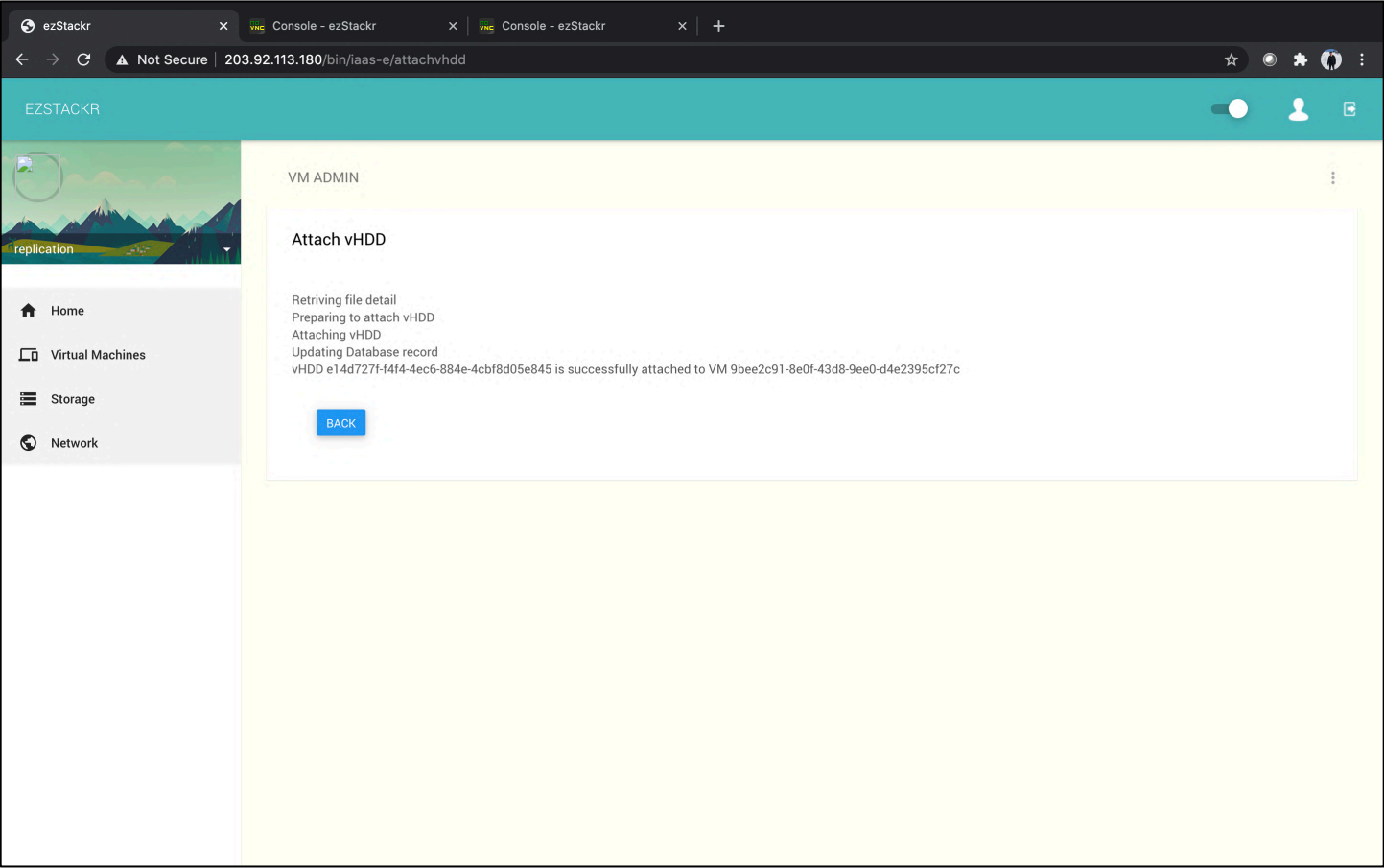
As User: Select “ez-u101” and “Attach Harddisk”



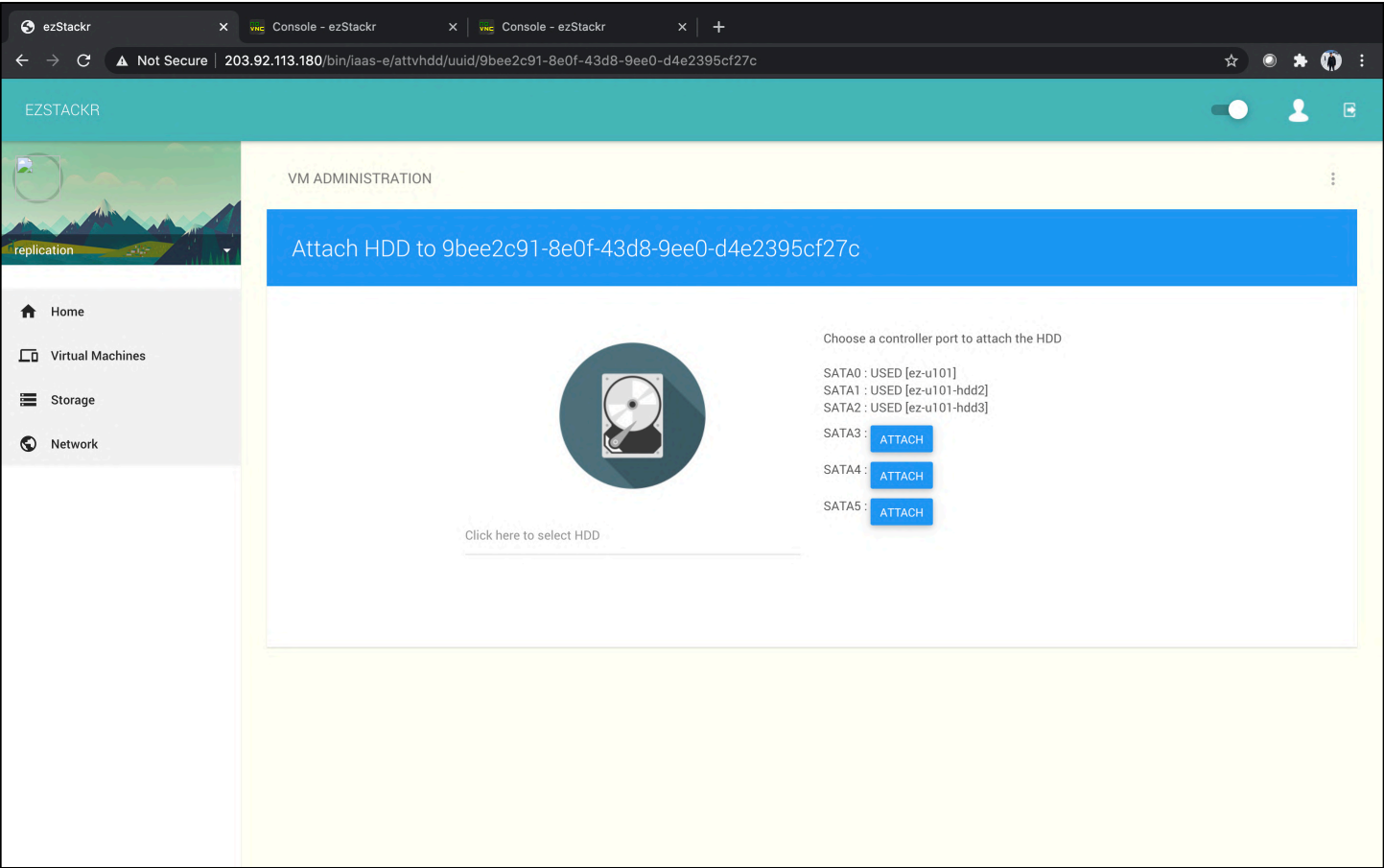
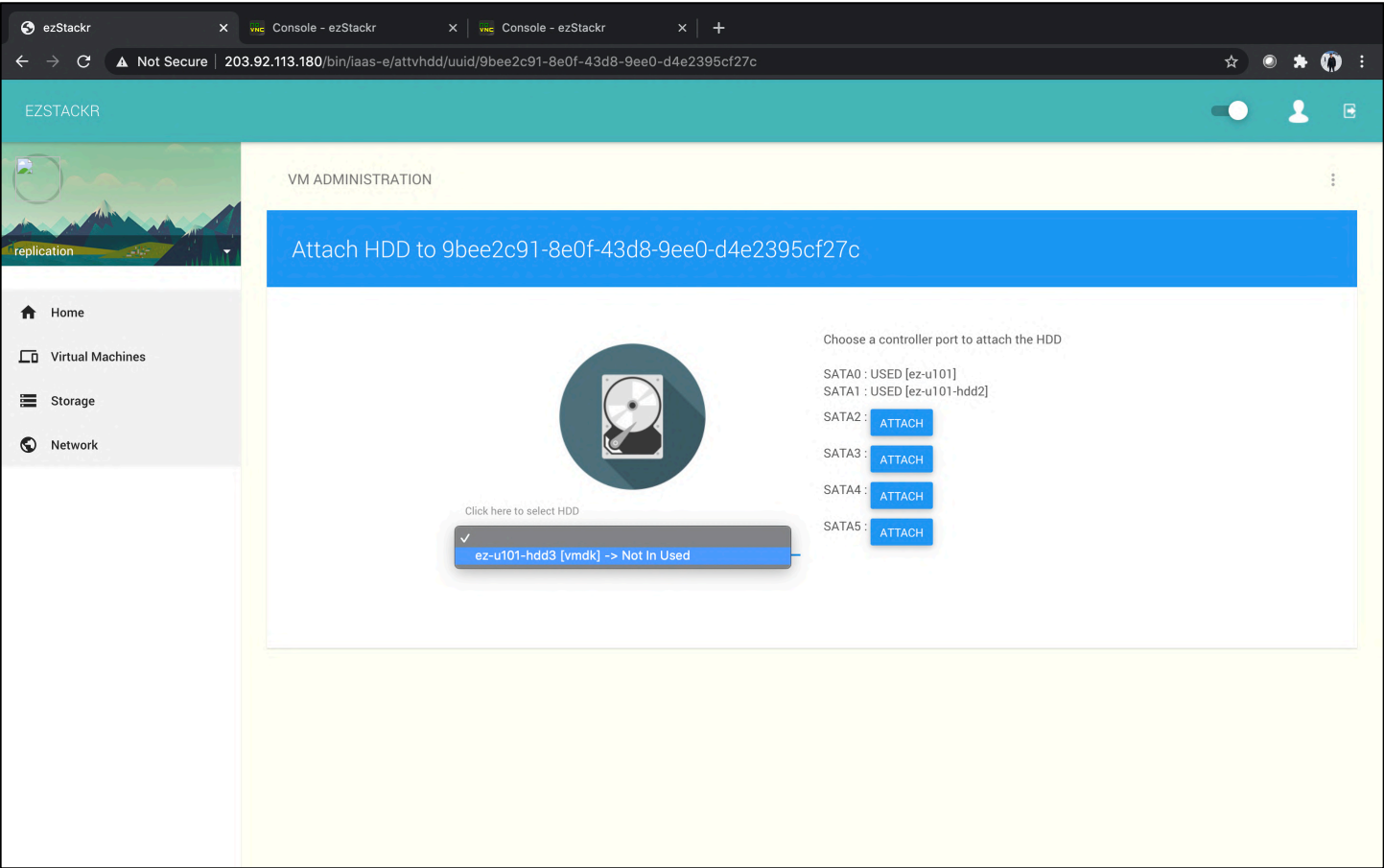
As User: Select “Click here to select HDD” and select “ez-u101-hdd2”



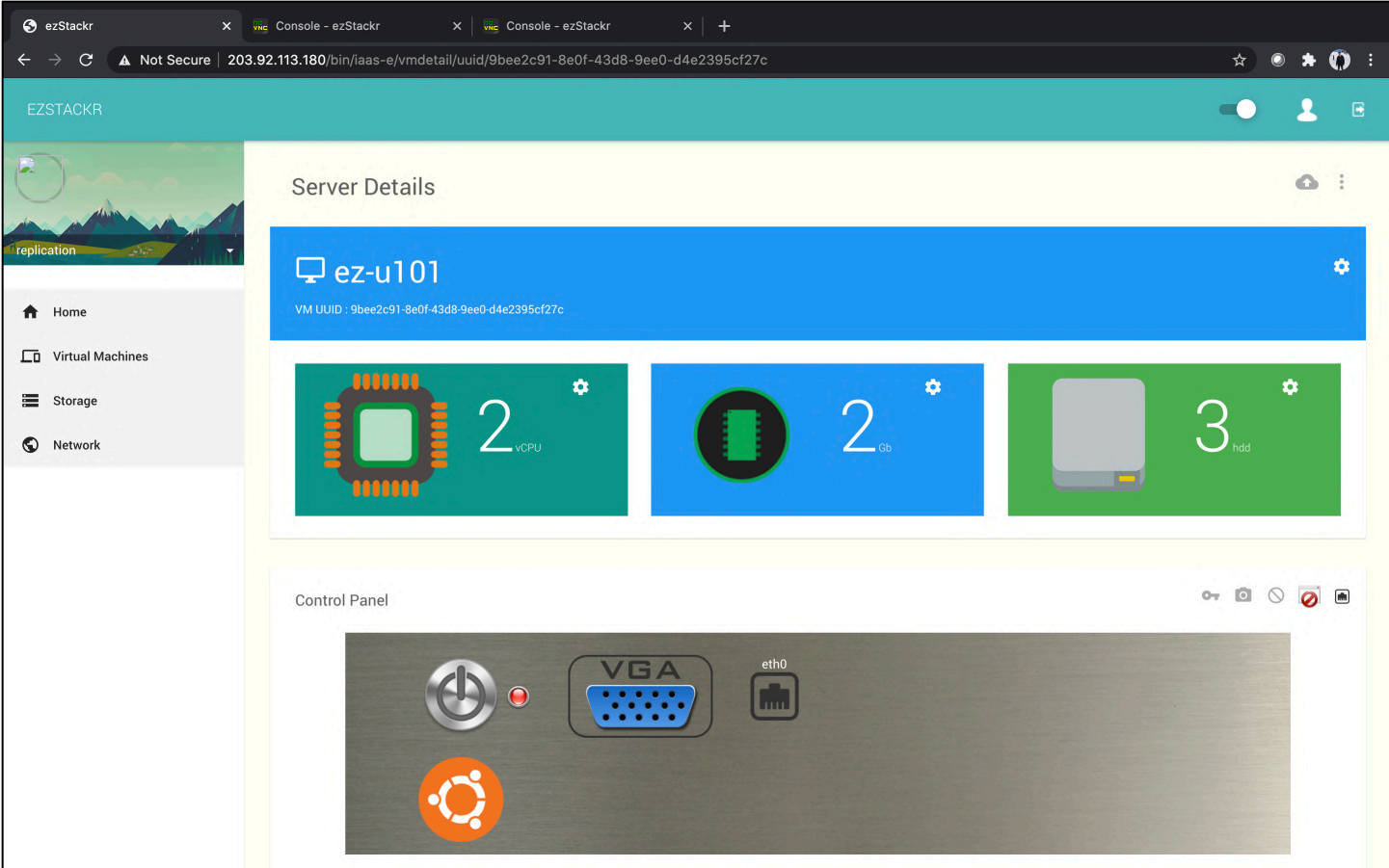
As User: Select “Attach”



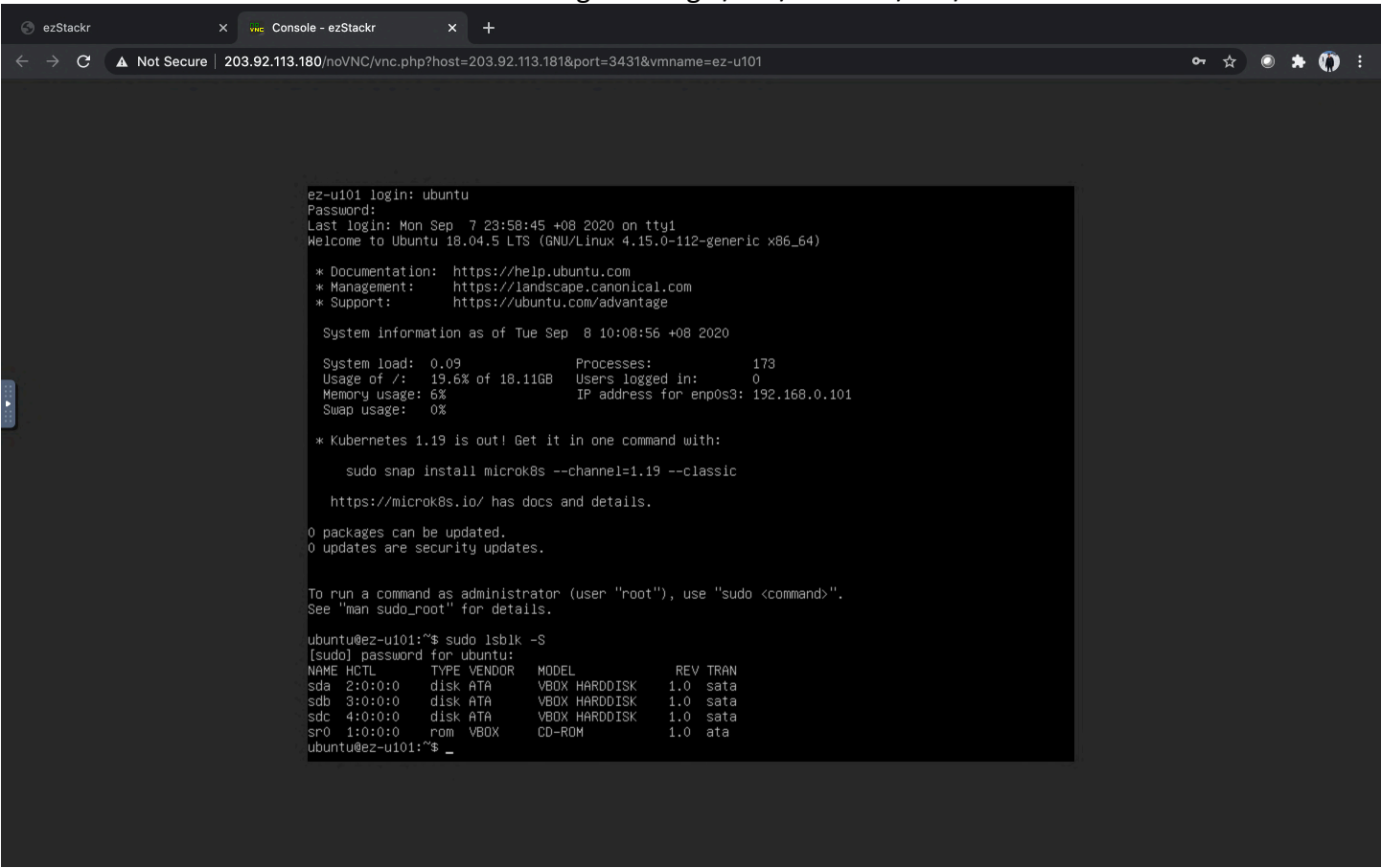
As User: Add the next harddisk



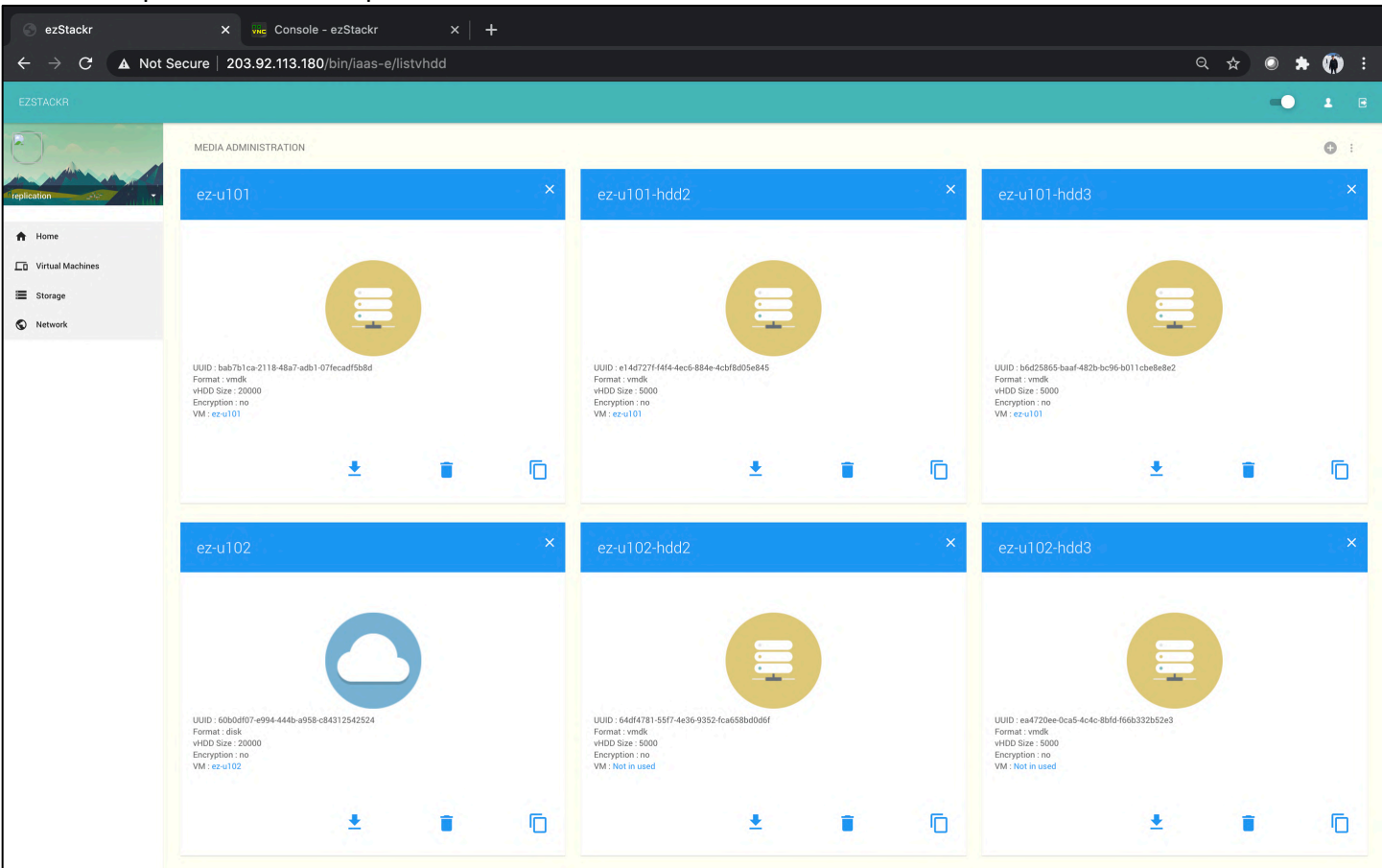
As User: View the VM details. There are 3 harddisks. Start the source VM.



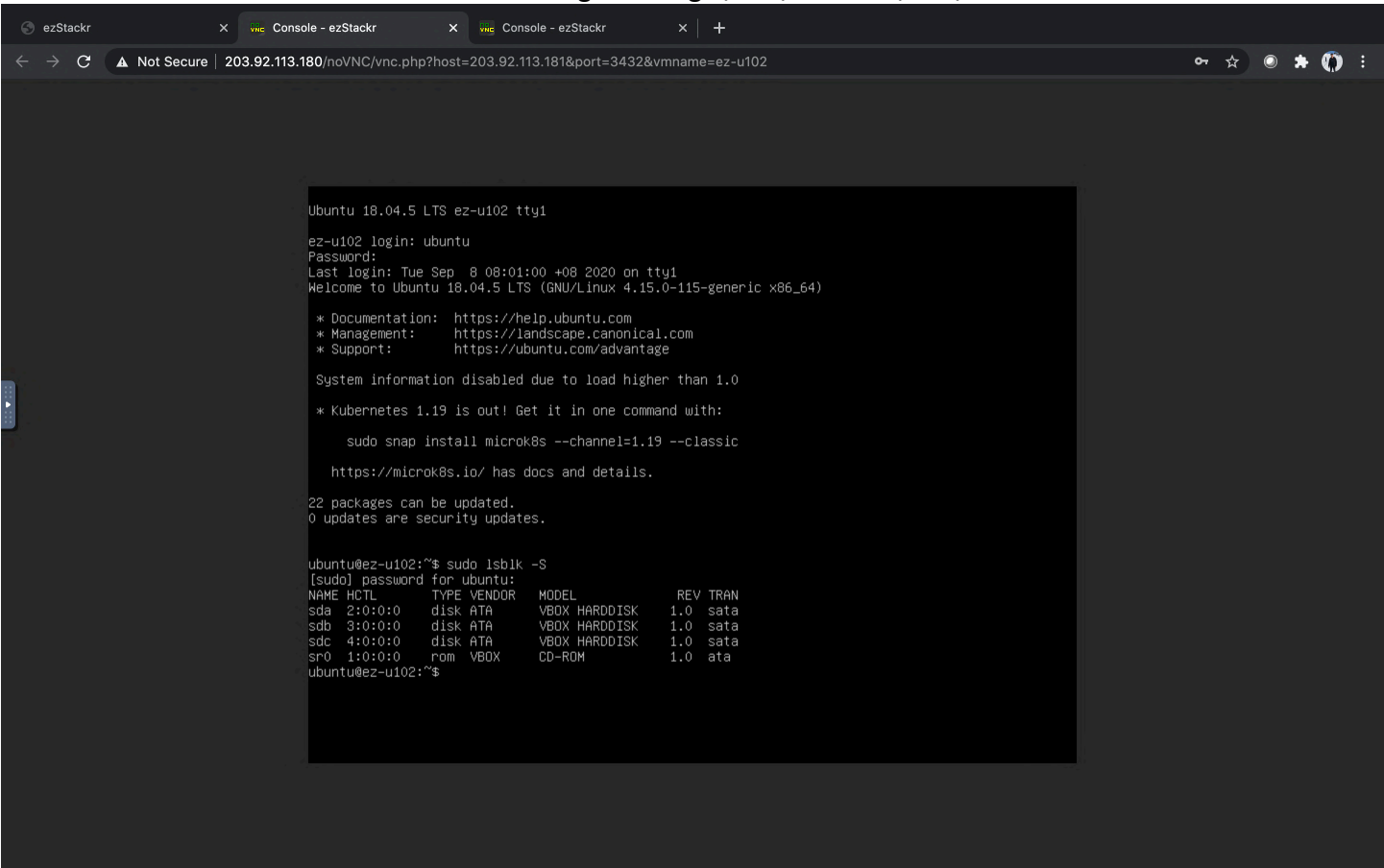
As User: “sudo lsblk -S” to view attached storage. Storage /dev/sdb and /dev/sdc has been attached



As User: Repeat the same steps for Destination VM



As User: “sudo lsblk -S” to view attached storage. Storage /dev/sdb and /dev/sdc has been attached



INSTALLING AND CONFIGURING ZFS FOR SOURCE VM AND DESTINATION VM

As User: Do the following steps on Source VM

Enable Ubuntu repository:

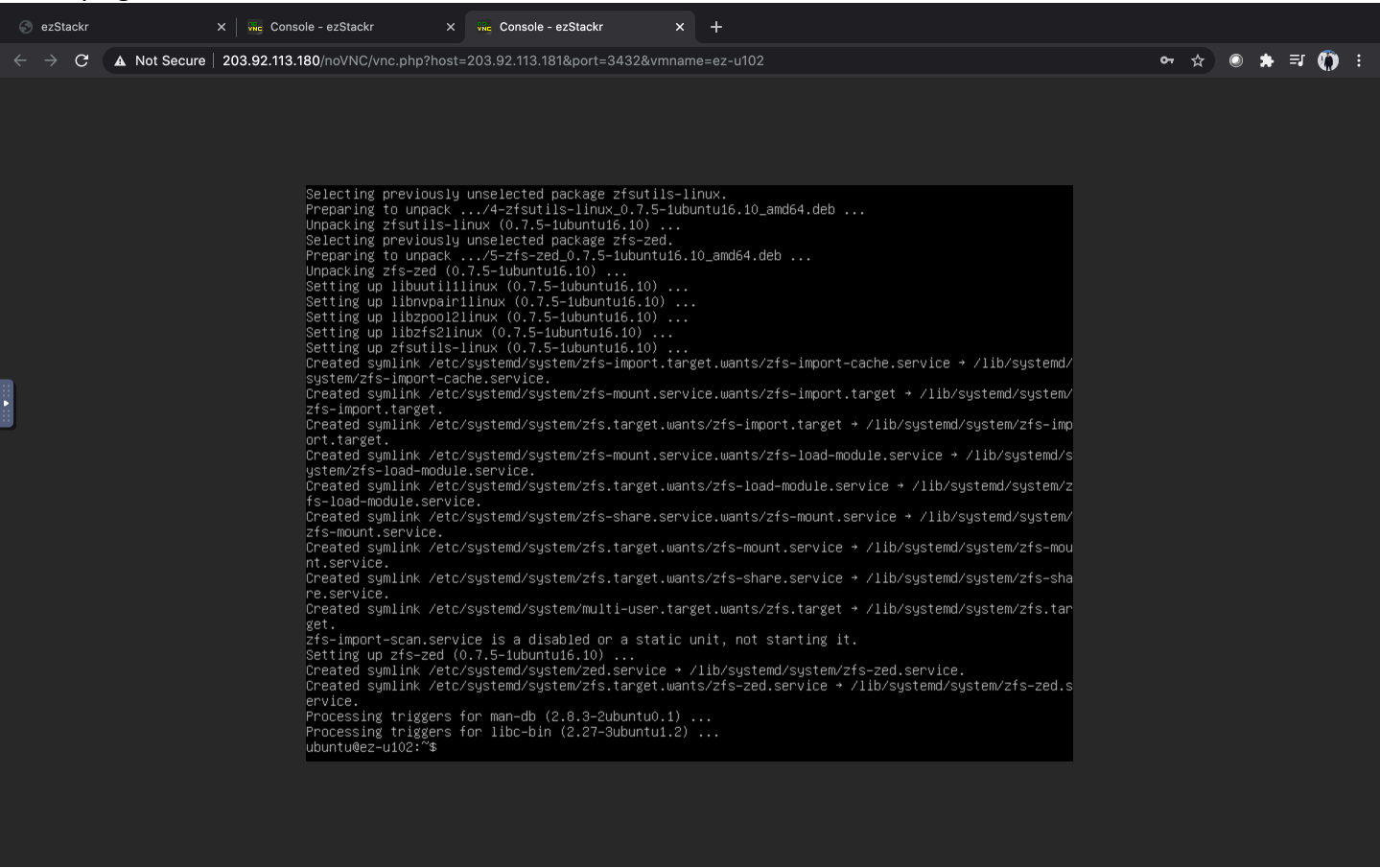
sudo apt-add-repository main
sudo apt-add-repository restricted
sudo apt-add-repository universe
sudo apt-add-repository multiverse

Update apt package:

sudo apt-get update

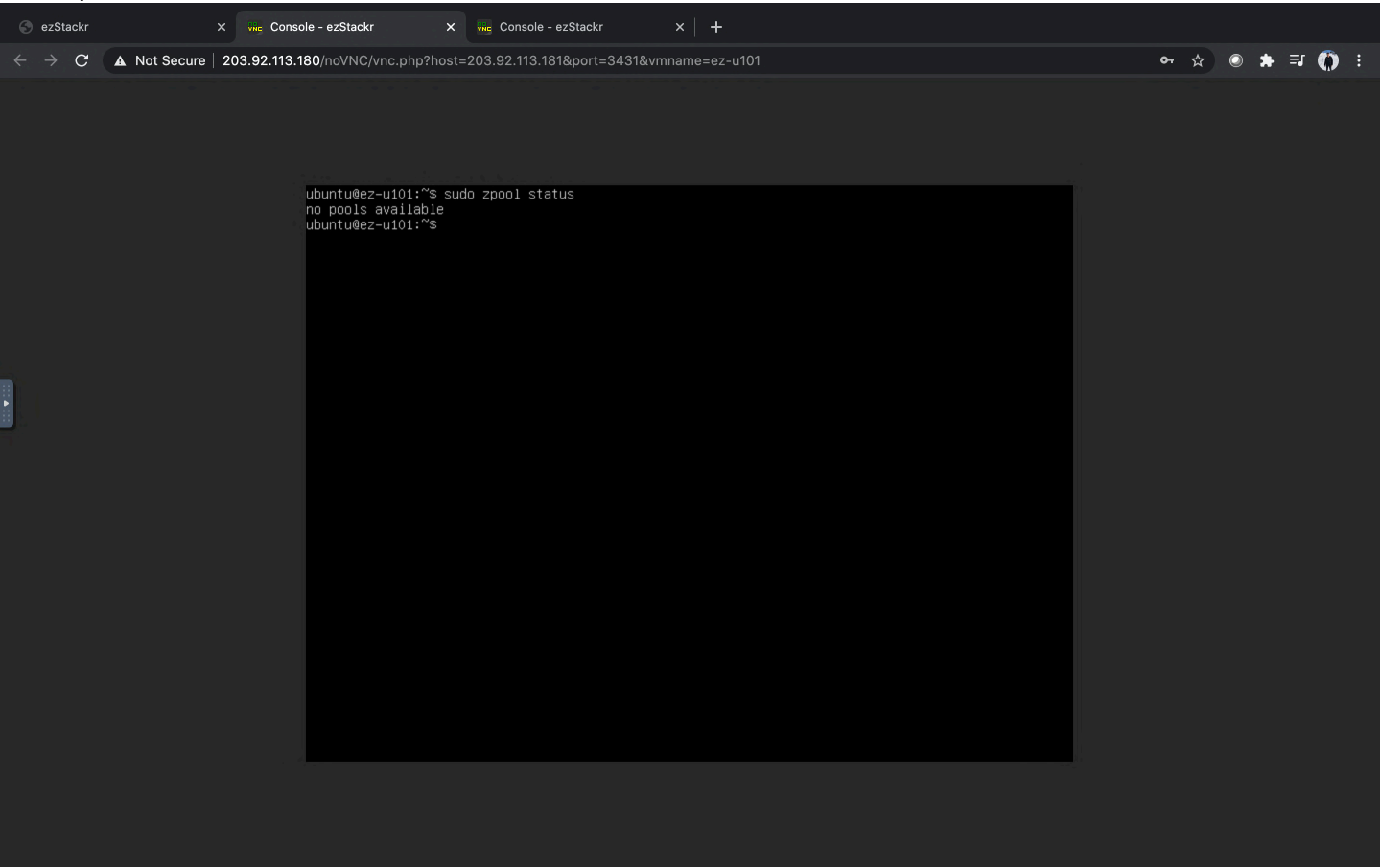
Install zfs:

sudo apt-get install zfsutils-linux



Check for any zfs pools:

sudo zpool status

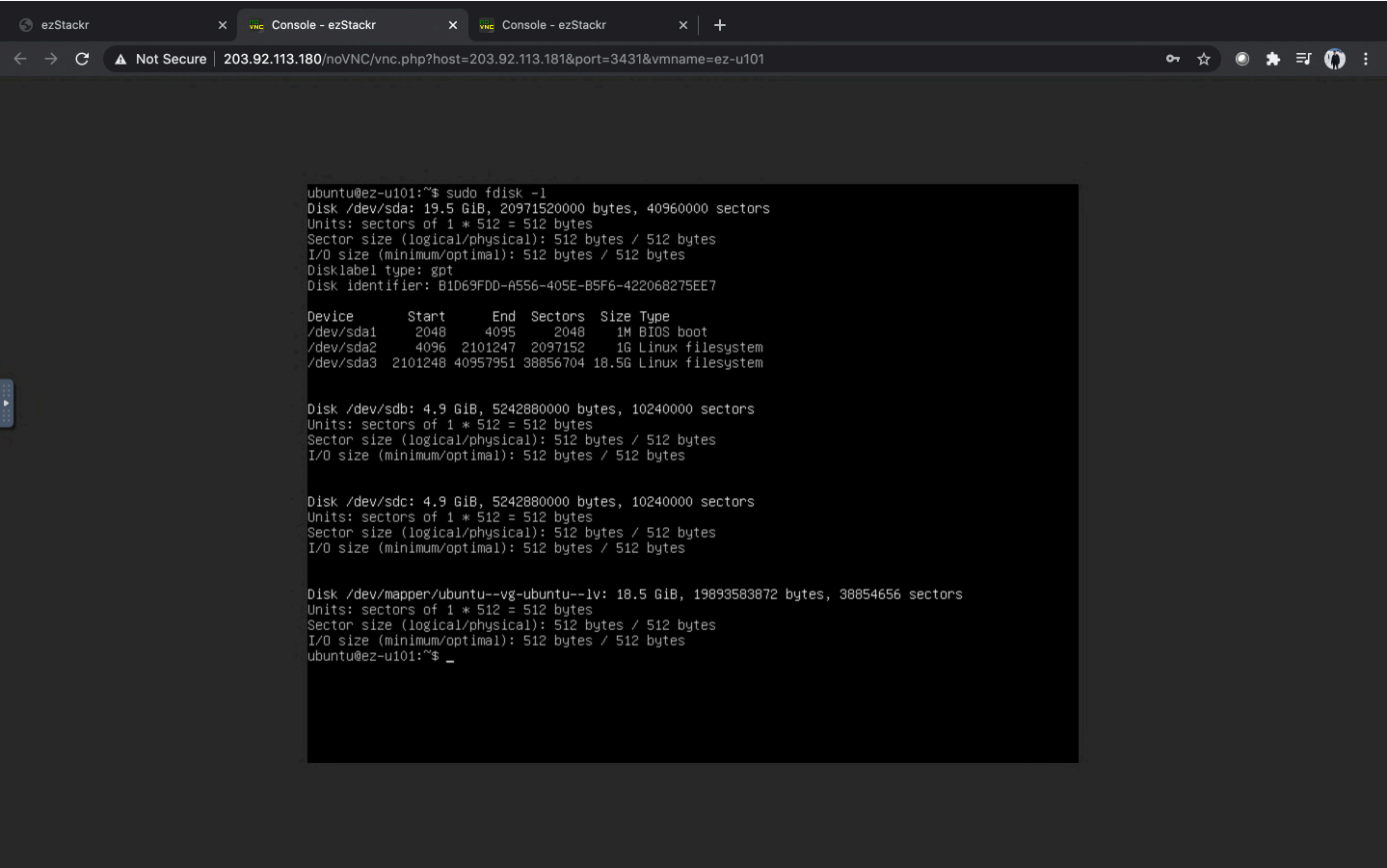


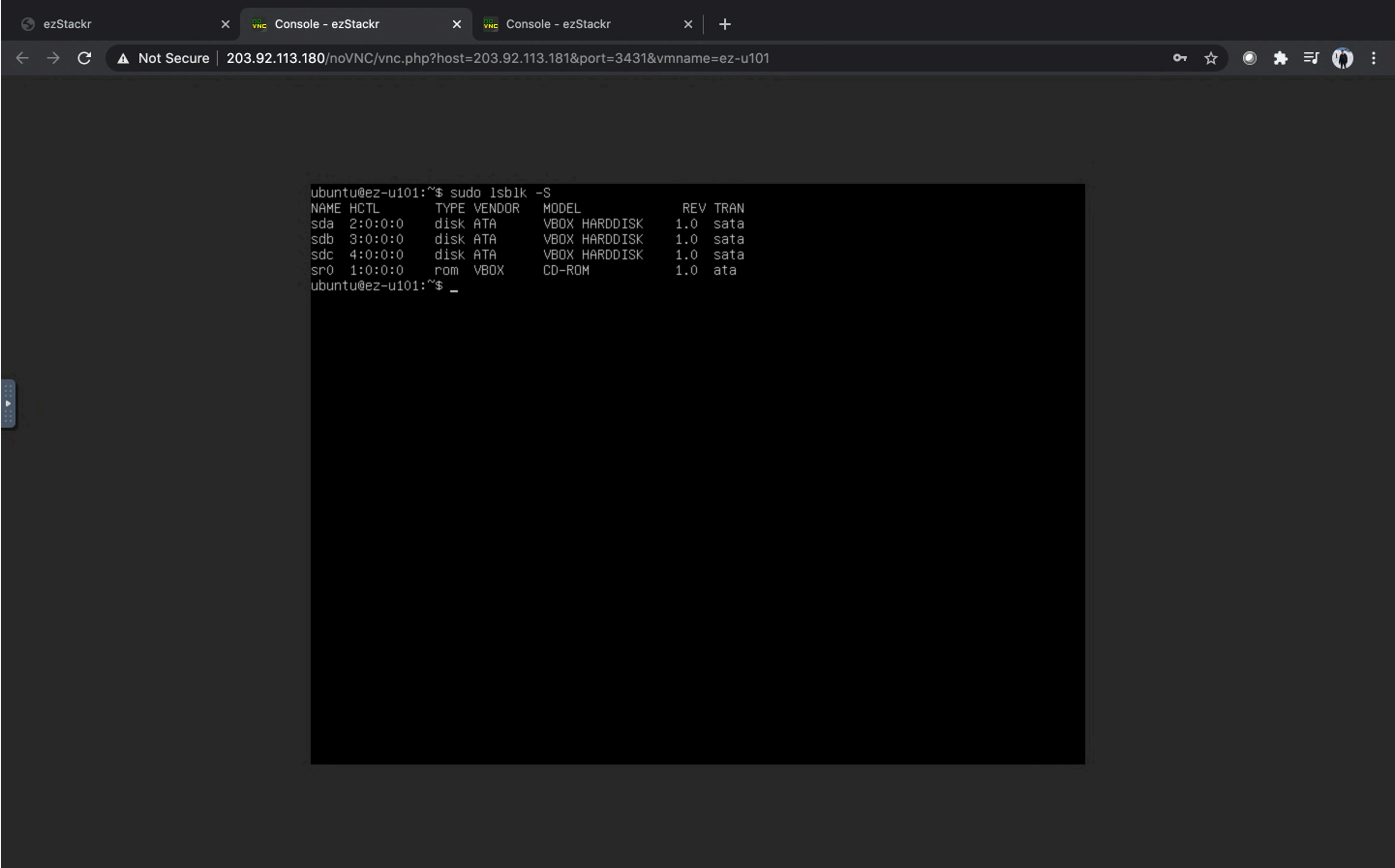
Display disk space:

df -H

List devices:

sudo fdisk -l
or
sudo lsblk -S





Create pool on source VM:
sudo zpool create -f files /dev/sdb /dev/sdc

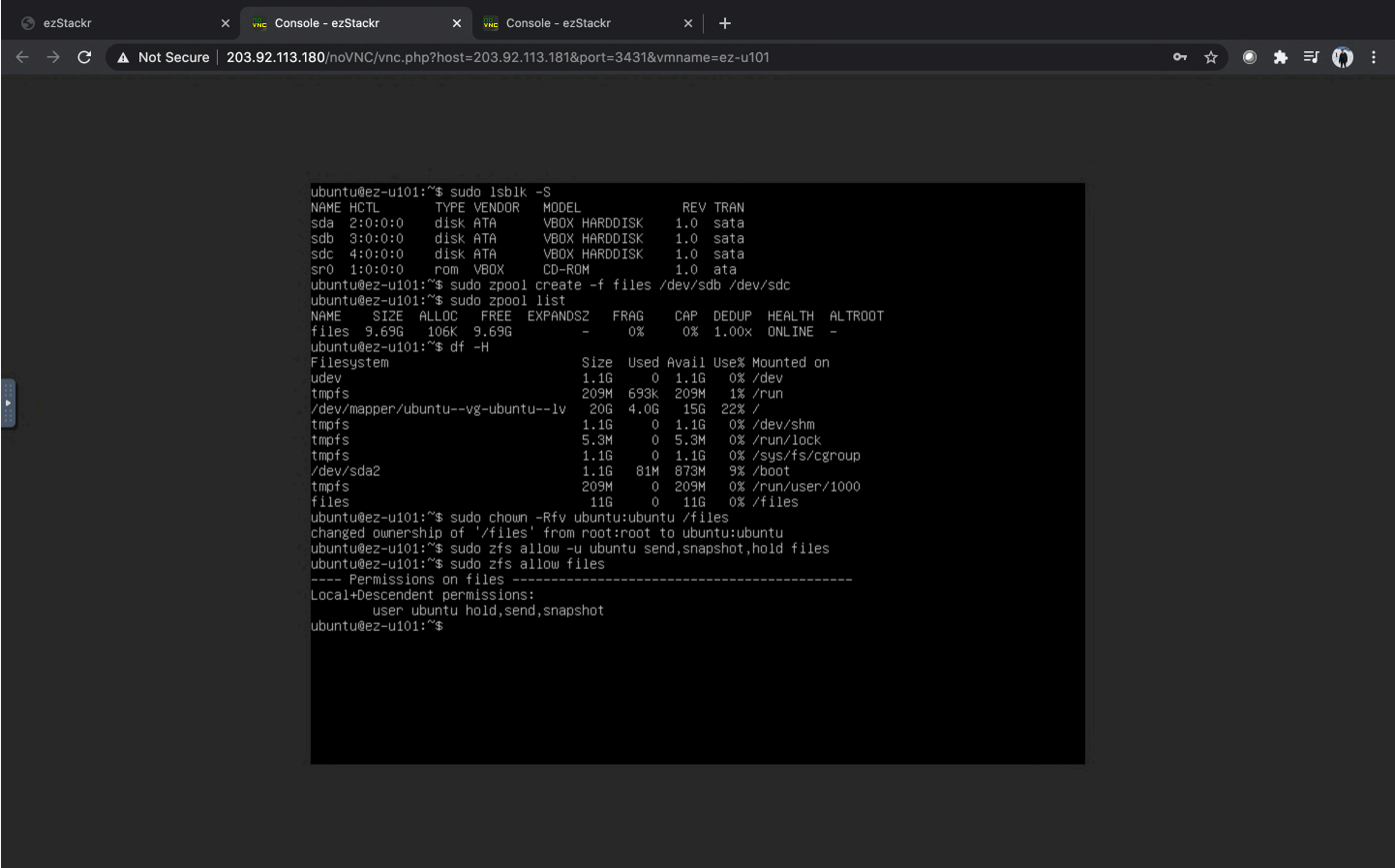
List pool on source VM:
sudo zpool list

Notice ZFS pool is mounted on /files automatically:
df -H

Change ownership on source VM:
sudo chown -Rfv ubuntu:ubuntu /files

Set permissions on source VM:
sudo zfs allow -u ubuntu send,snapshot,hold files

List permissions on source VM:
sudo zfs allow files



As User: Do the following steps on Destination VM

Enable Ubuntu repository:
sudo apt-add-repository main
sudo apt-add-repository restricted
sudo apt-add-repository universe
sudo apt-add-repository multiverse

Update apt package:
sudo apt-get update

Install zfs:
sudo apt-get install zfsutils-linux

Check for any zfs pools:
sudo zpool status

Display disk space:
df -H

List devices:
sudo fdisk -l
or
sudo lsblk -S

Create pool on destination VM:
sudo zpool create -f filesbkup /dev/sdb /dev/sdc

List pool on destination VM:
sudo zpool list

ZFS pool is mounted on /filesbkup automatically:
df -H

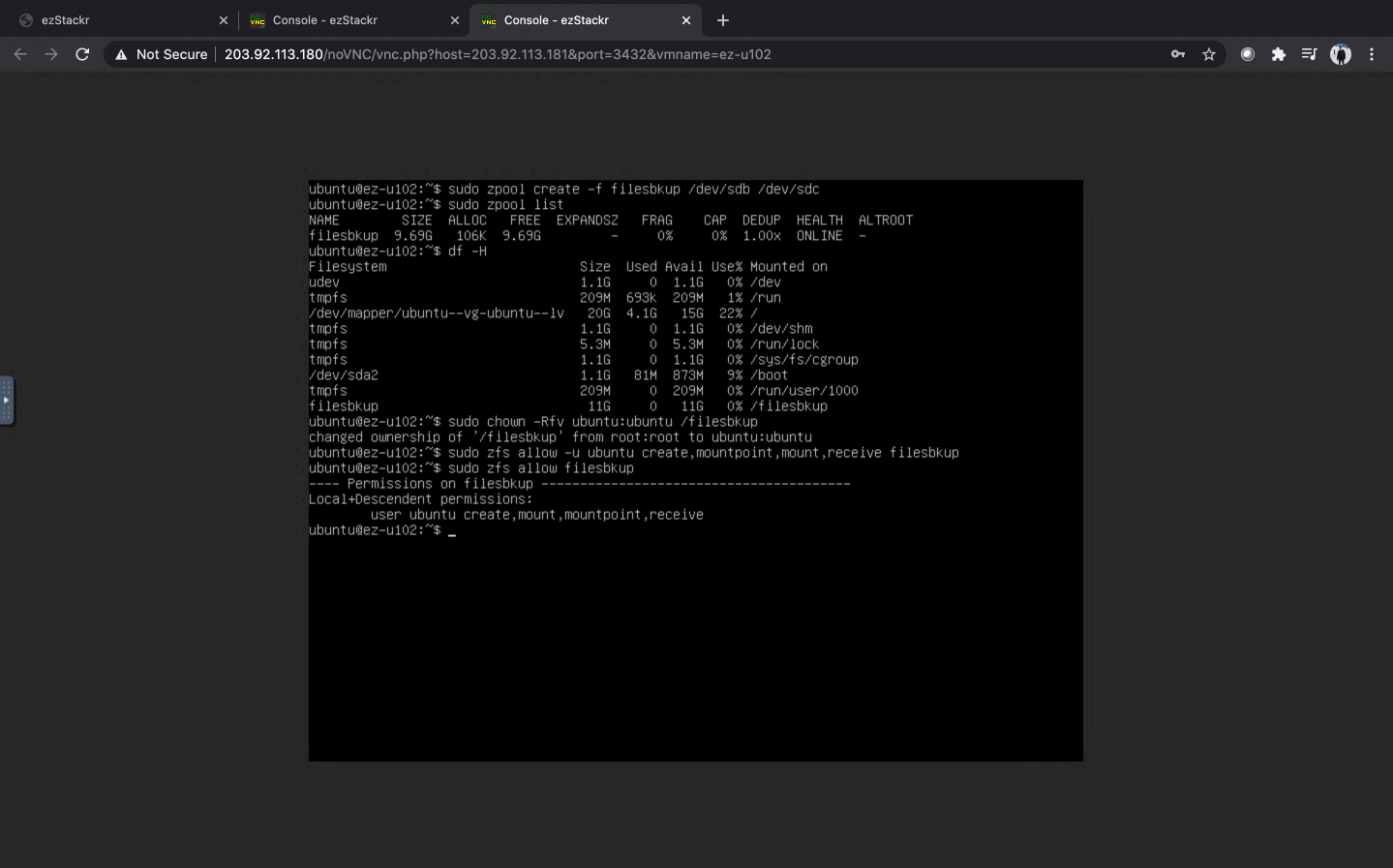
Change ownership on destination VM:
sudo chown -Rfv ubuntu:ubuntu /filesbkup

Set permissions on destination VM:

sudo zfs allow -u ubuntu create,mountpoint,mount,receive filesbkup

List permissions on destination VM:

sudo zfs allow filesbkup



INSTALLING AND CONFIGURING OPENSSH FOR SOURCE VM AND DESTINATION VM

As User: Do the following steps on Source VM and Destination VM

S

Install ssh:

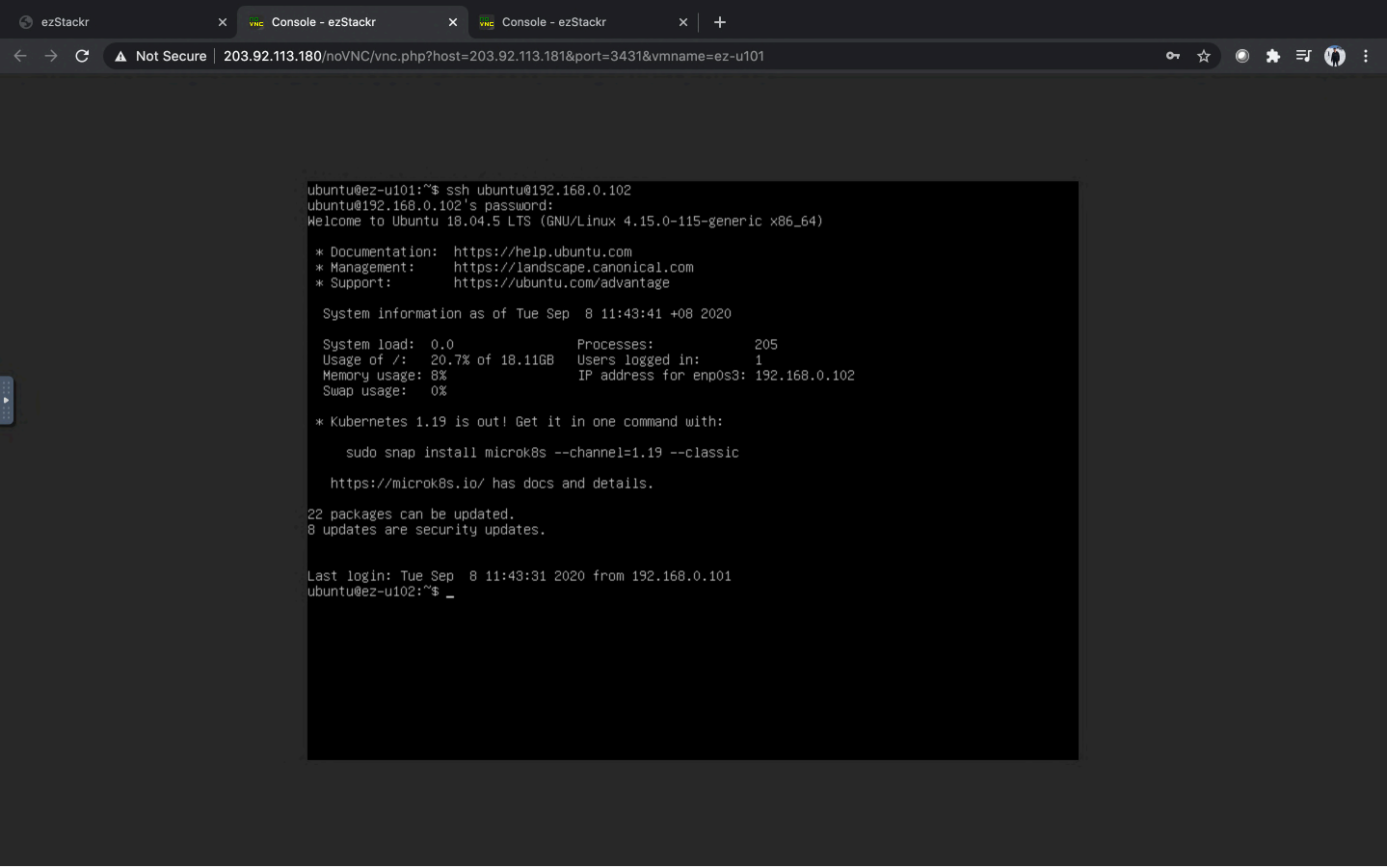
sudo apt install openssh-server

Allow ssh on ufw:

sudo ufw allow ssh

Test ssh login with password at the source VM, login to destination VM:

ssh ubuntu@192.168.0.102



Test ssh login with password at the destination VM, login to source VM:

ssh ubuntu@192.168.0.101

CONFIGURING OPENSSH FOR SOURCE VM AND DESTINATION VM TO LOGIN WITHOUT PASSWORDS

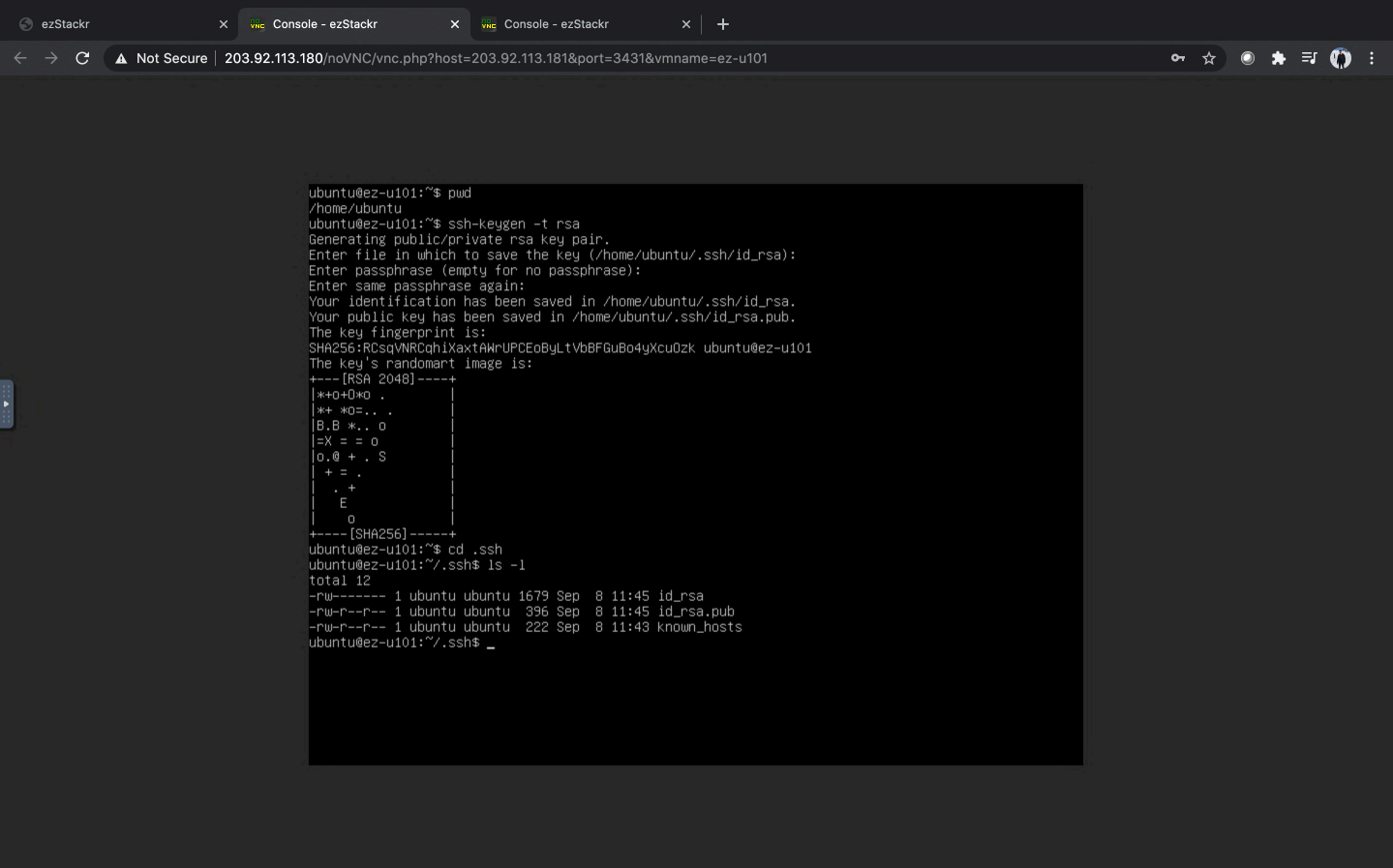
Setup ssh login without password on source VM:

cd \$HOME

ssh-keygen -t rsa

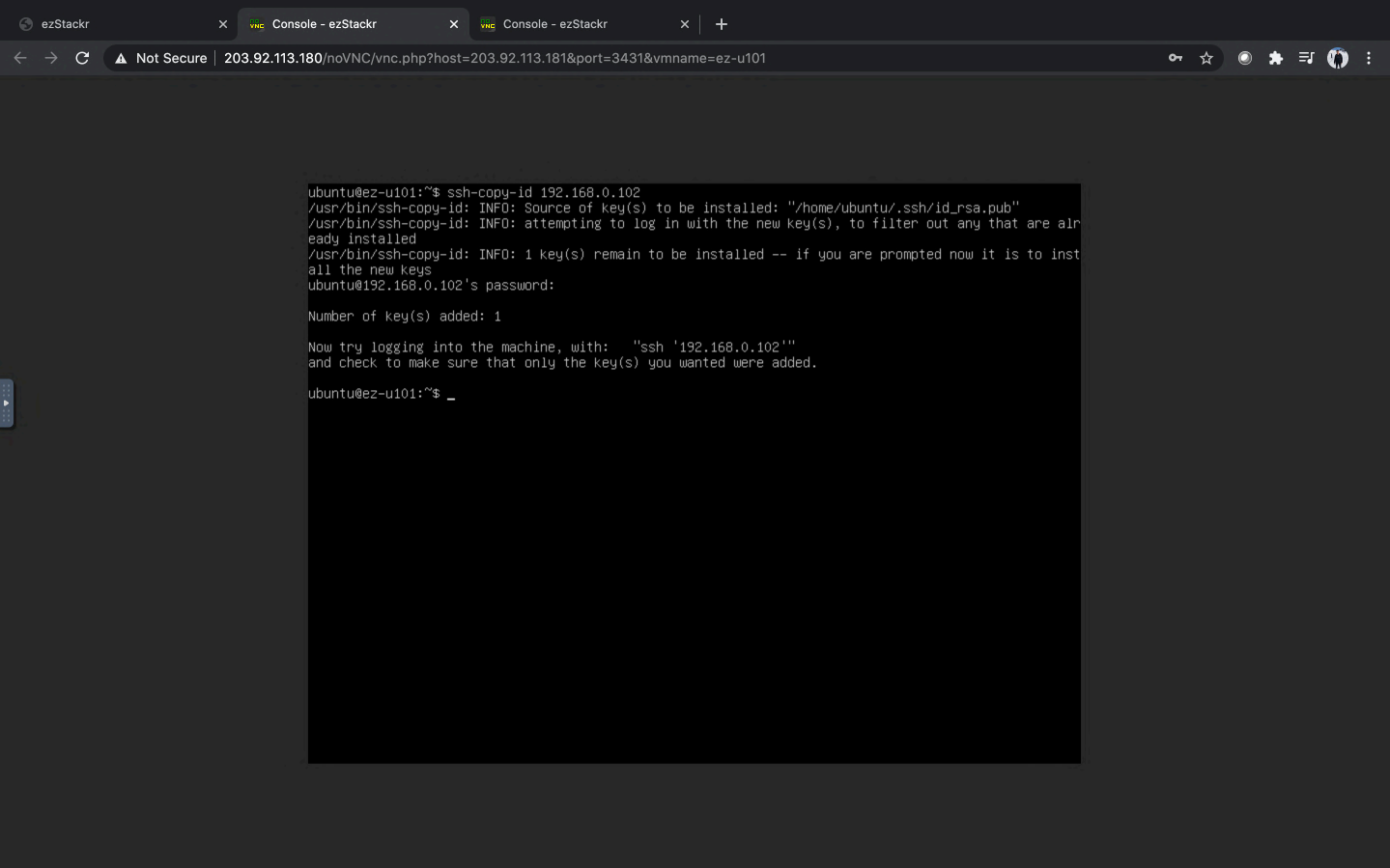
cd ~/.ssh

ls -l

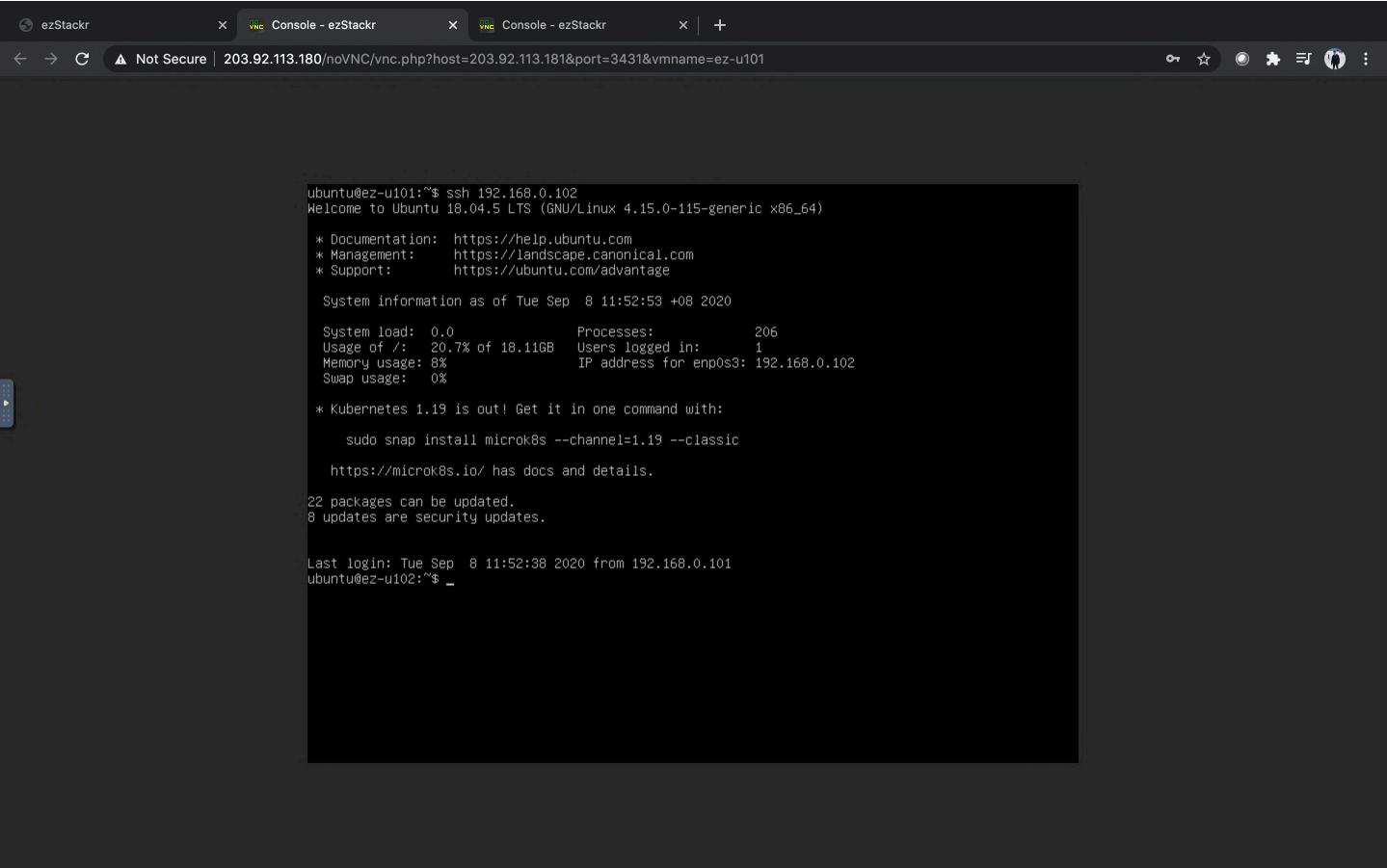


Transfer public key to destination. This will start a ssh session, which you will need to authenticate with your password. After you enter your password, it will copy your public key to the destination’s authorized keys file, which will allow you to log in without the password next time. At the source VM:

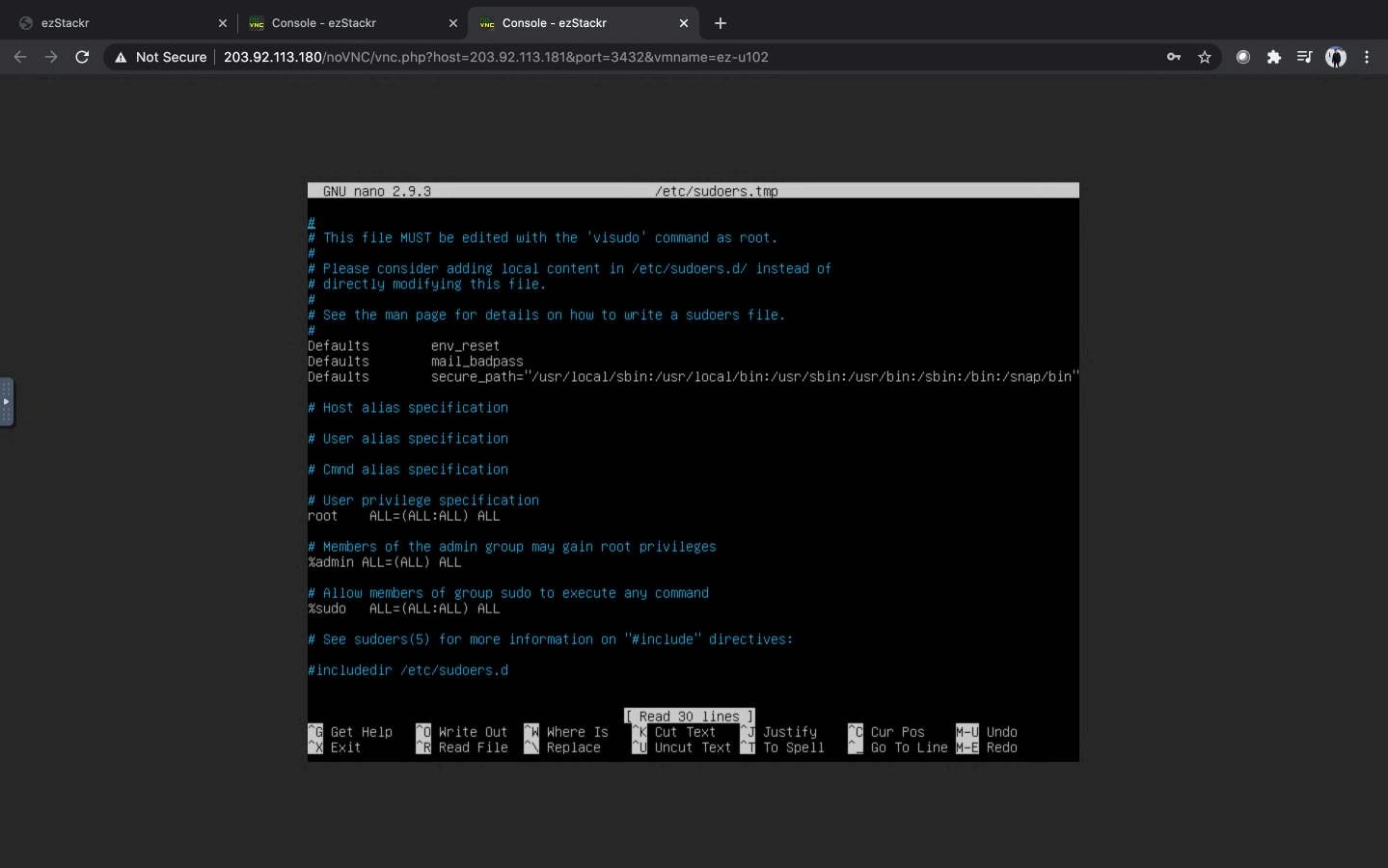
ssh-copy-id 192.168.0.102



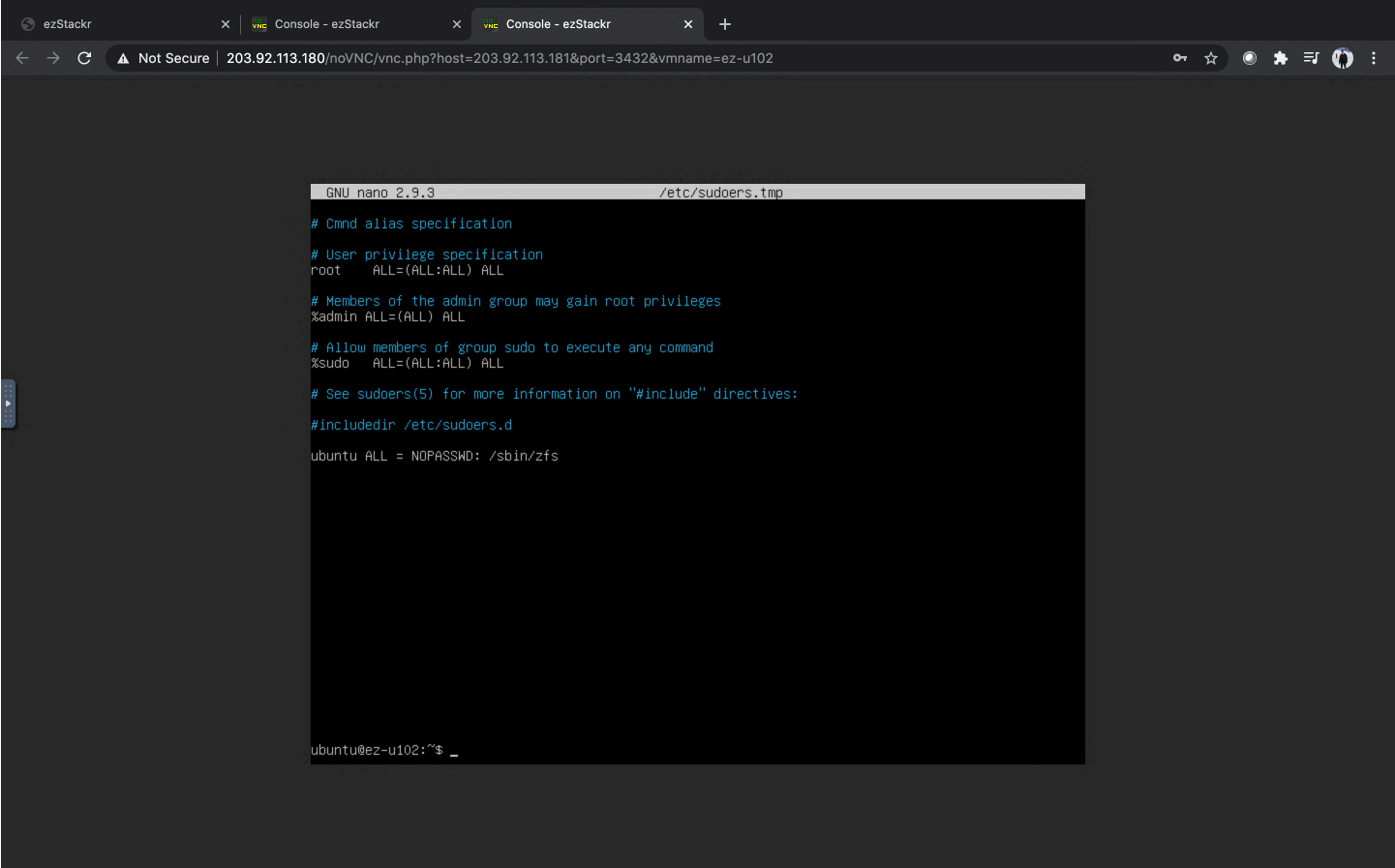
Test automated login from source VM to destination VM:
ssh 192.168.0.102



Allow destination VM to run zfs without prompting a password when destination executes "sudo zfs". If you do not do this step, there will be an error message "no tty present and no askpass program specified". Do this at the destination VM:
sudo visudo



Goto end of file and add and save the file:
ubuntu ALL = NOPASSWD: /sbin/zfs



CREATING SNAPSHOTS ON SOURCE VM

Create first snapshot:
sudo zfs snapshot files@snapshot1

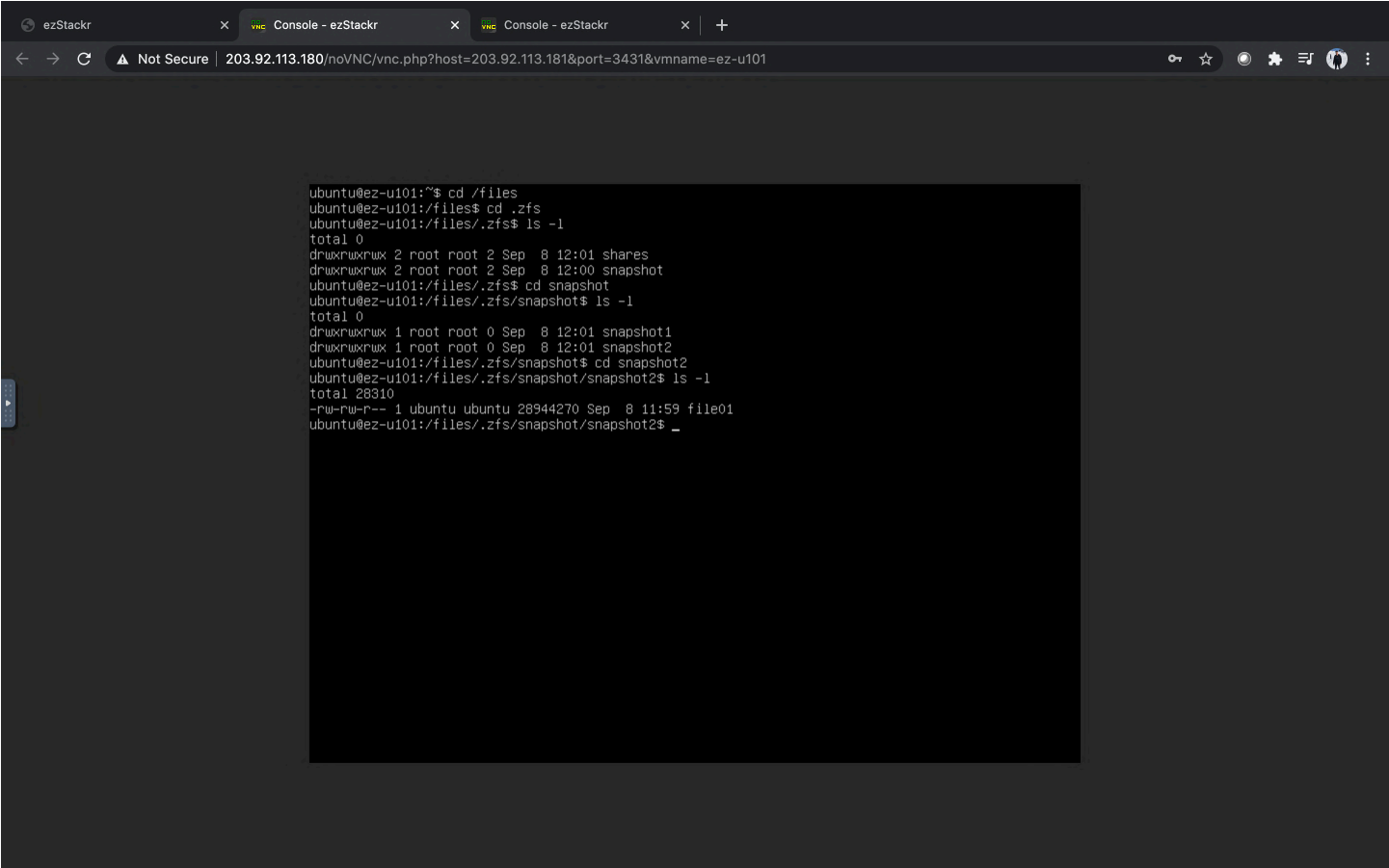
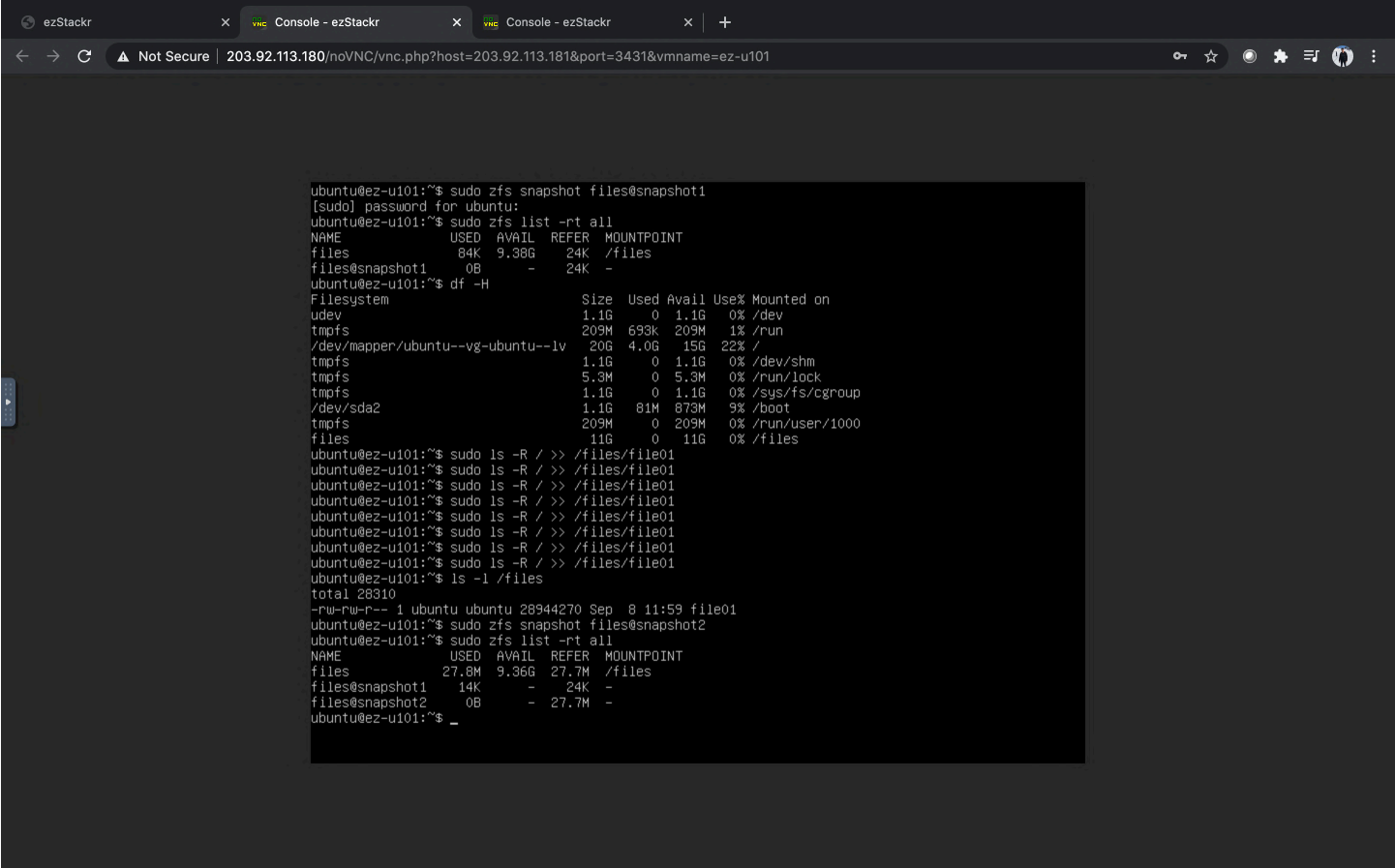
List snapshot:
sudo zfs list -rt all

Add a file into /files:
e.g. sudo ls -R / >> /files/file01
do this several times to expand the size of /files/file01

List the size of /files/file01:
ls -l /files

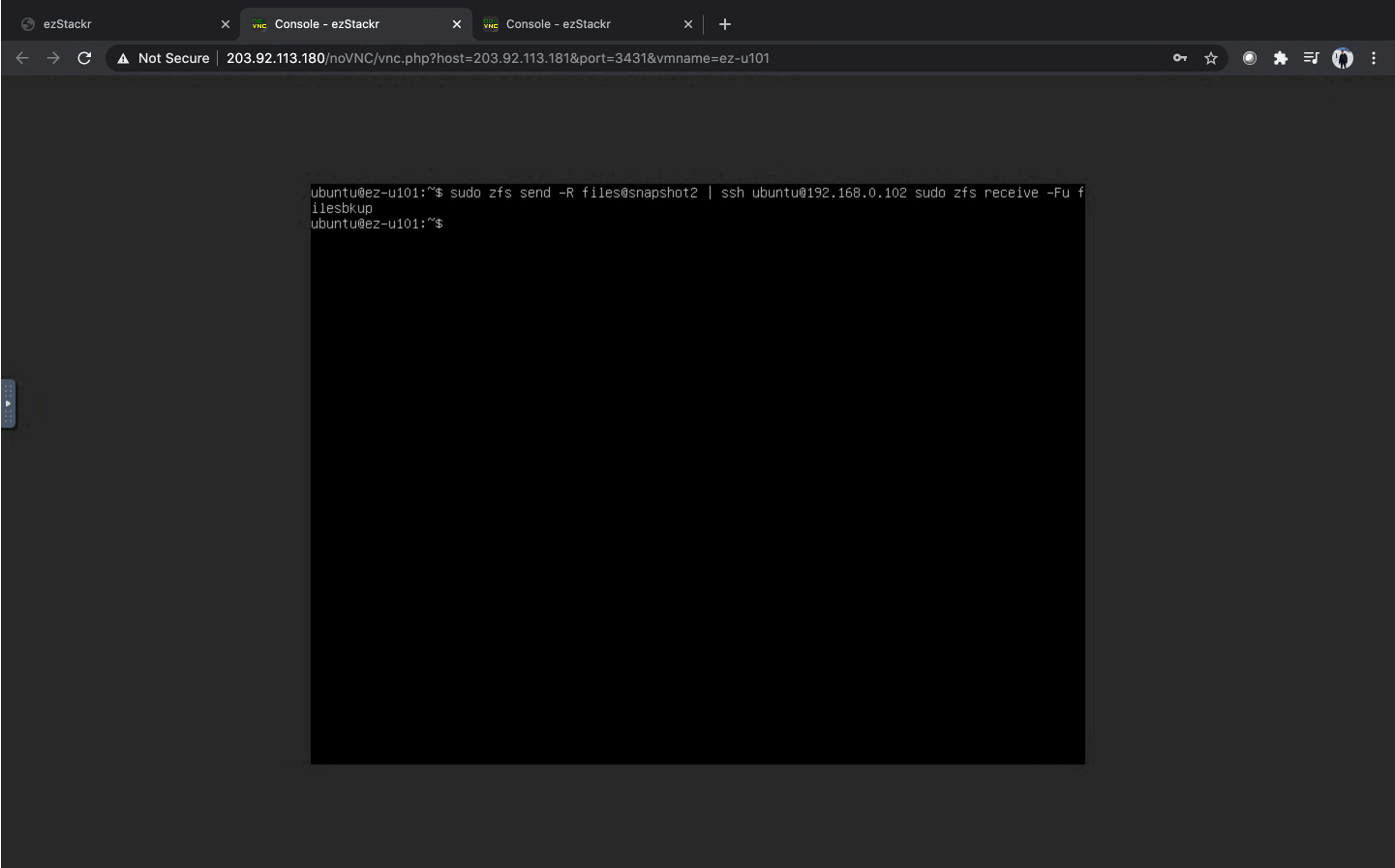
Create second snapshot:
sudo zfs snapshot files@snapshot2

List snapshot:
sudo zfs list -rt all



REPLICATING SNAPSHOTS ON SOURCE VM TO DESTINATION VM

Replicate the snapshot:
sudo zfs send -R files@snapshot2 | ssh ubuntu@192.168.0.102 sudo zfs receive -Fu filesbkup

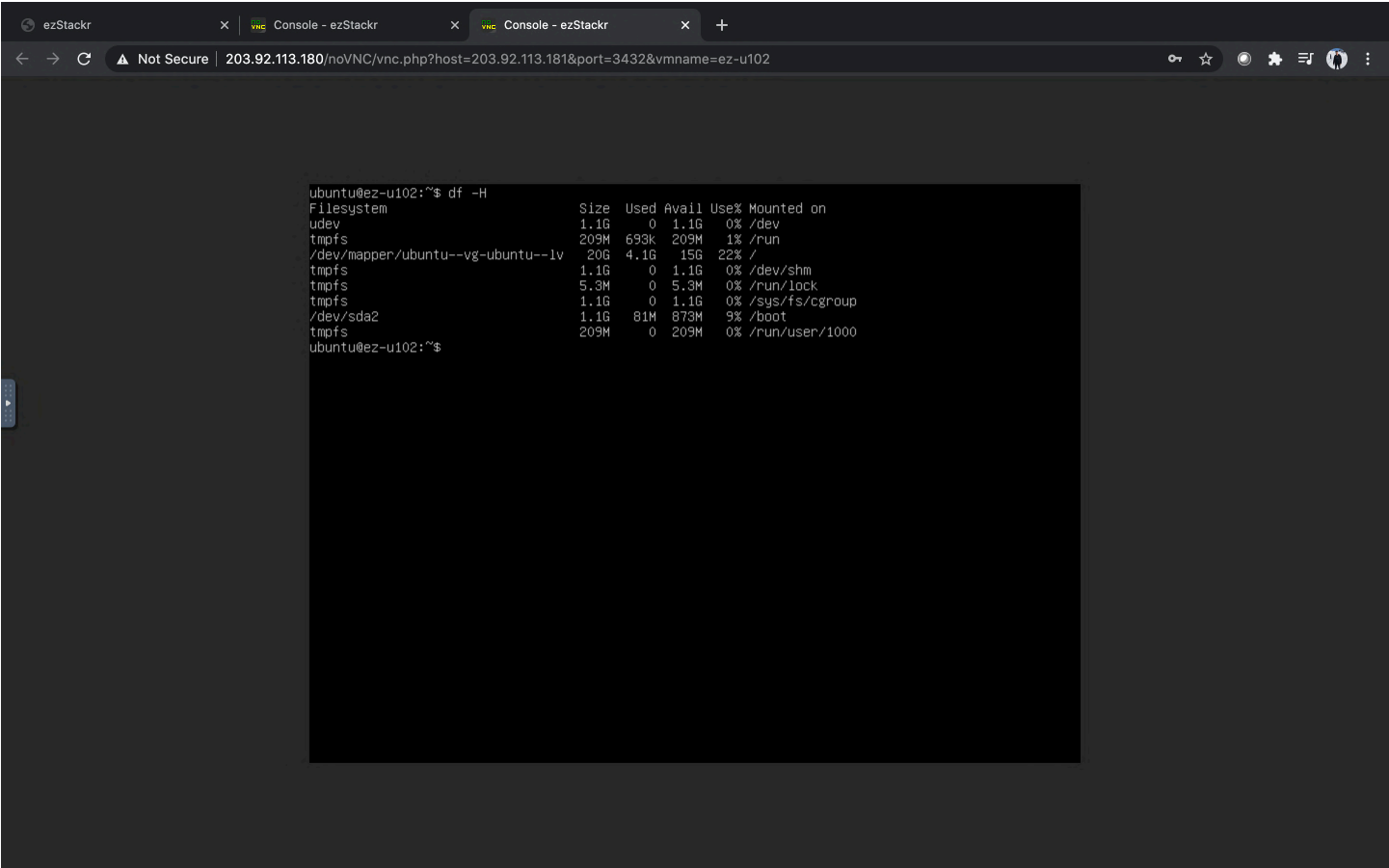


Zfs will unmount the destination VM pool and replicate the snapshot:

CHECKING REPLICATED SNAPSHOTS ON DESTINATION VM

Zfs will unmount the destination VM pool and replicate the snapshot. Verify that destination VM’s /filesbkup was automatically unmounted:

df -H



Remount the destination VM pool file system:

sudo zfs mount filesbkup

List storage on destination VM to check if destination VM has received the data:

df -H

[u can see the storage used has now grown by the size of the snapshot on destination VM pool]

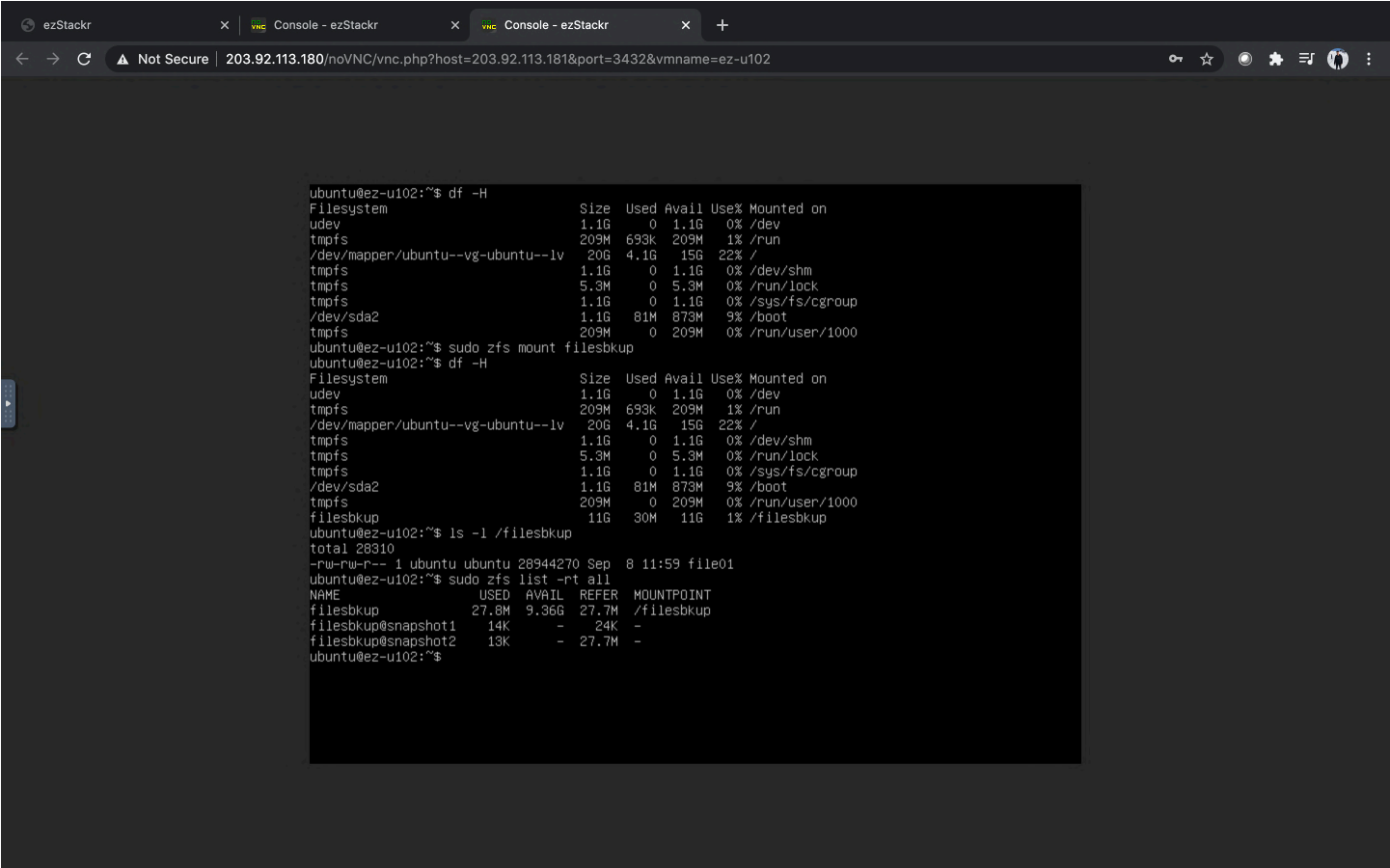
List the size of /filesbkup/file01:

ls -l /filesbkup

List snapshot on Destination VM:

sudo zfs list -rt all

[u can see the new snapshot on remote pool]



Conclusion:

Replication is successful from source VM to destination VM

SIMULATING REPLICATION AGAIN BY DESTROYING AND RECREATING THE POOL

Destroy the Destination VM’s pool:

sudo umount /filesbkup

sudo zpool destroy filesbkup

Recreate pool on destination VM:
sudo zpool create -f filesbkup /dev/sdb /dev/sdc

List pool on destination VM:
sudo zpool list

ZFS pool is mounted on /filesbkup automatically:
df -H

Change ownership on destination VM:
sudo chown -Rfv ubuntu:ubuntu /filesbkup

Set permissions on destination VM:
sudo zfs allow -u ubuntu create,mountpoint,mount,receive filesbkup

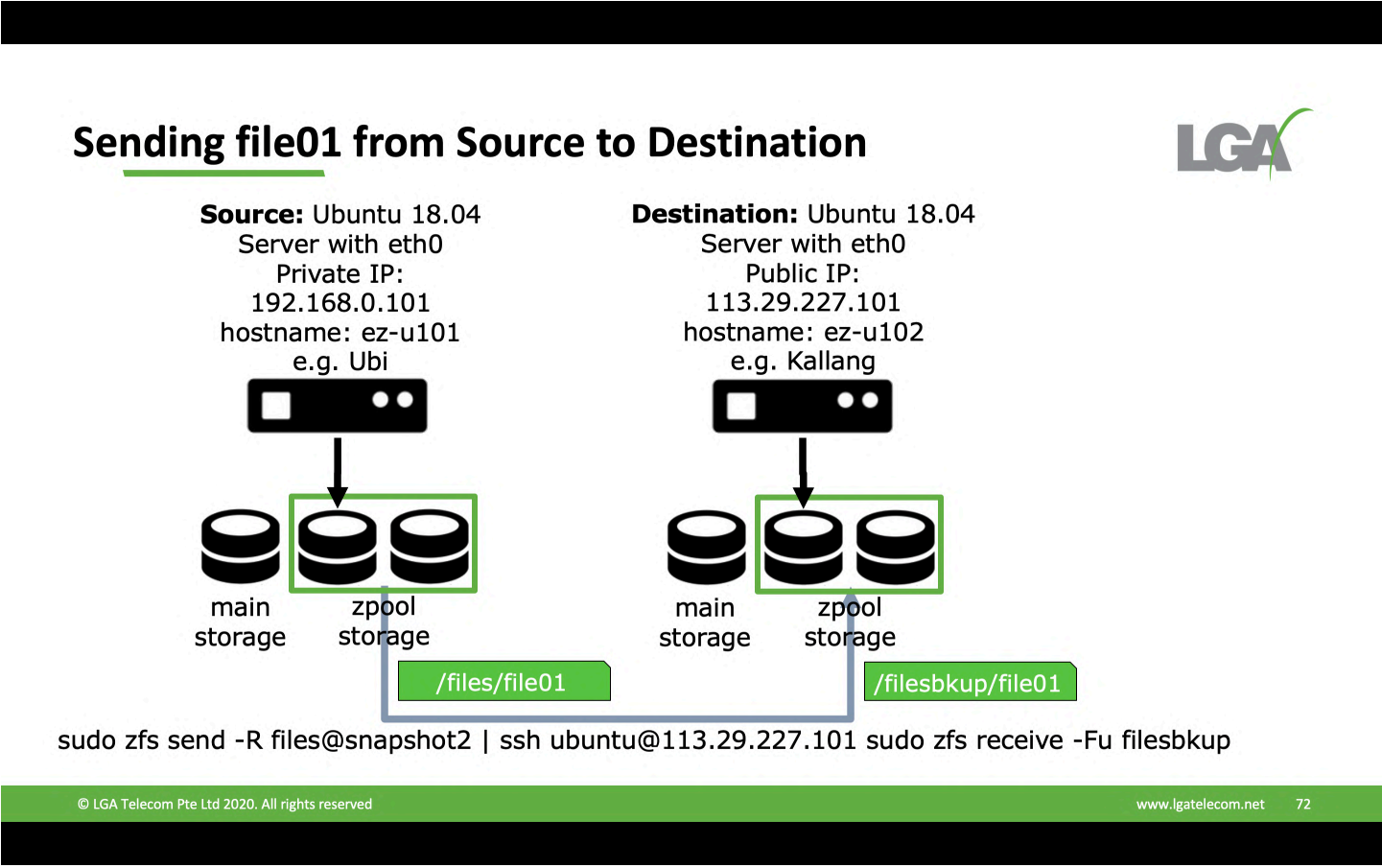
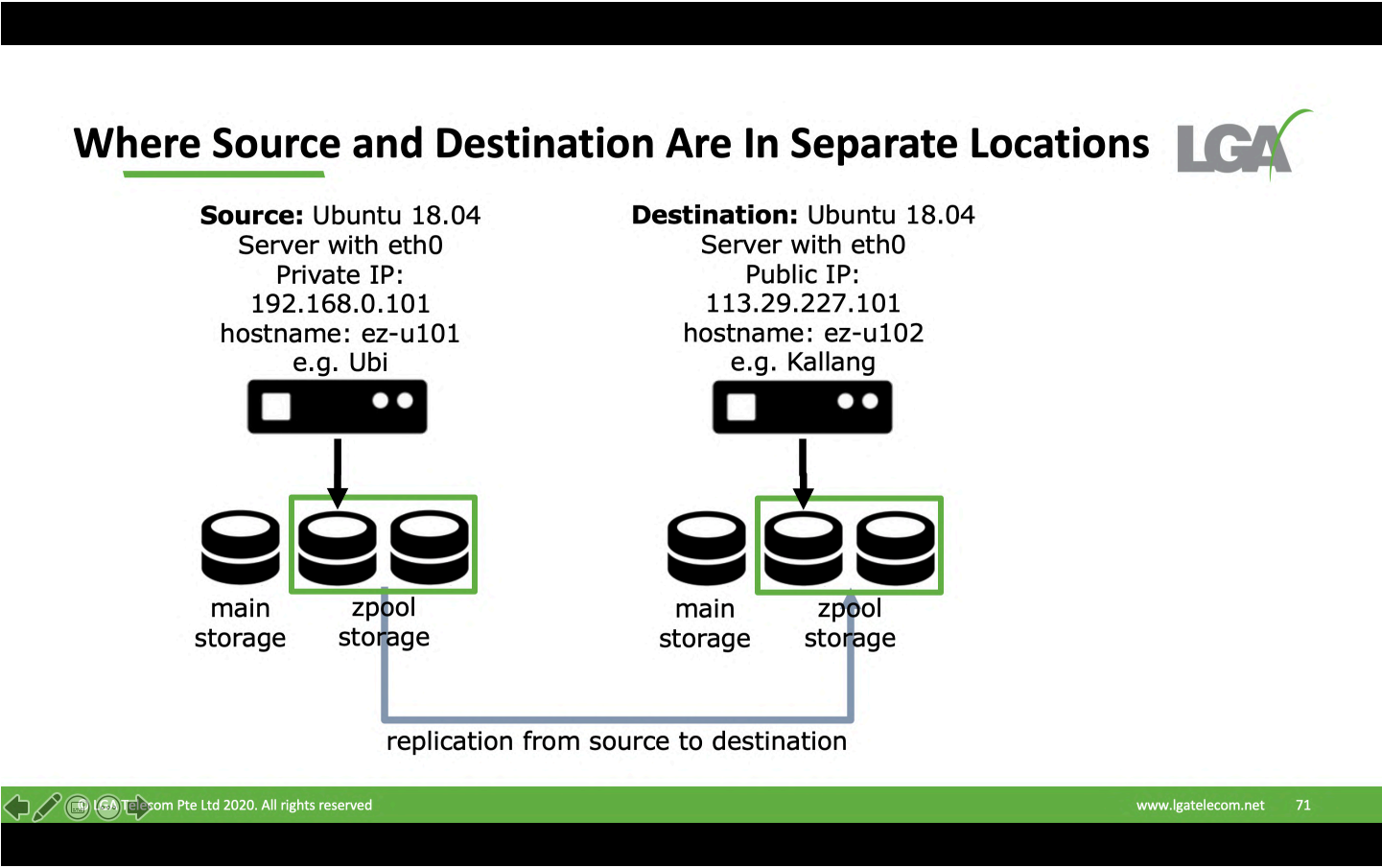
List permissions on destination VM:
sudo zfs allow filesbkup

OTHER COMMANDS
See where snapshots are kept:
cd /filesbkup
ls -l .zfs

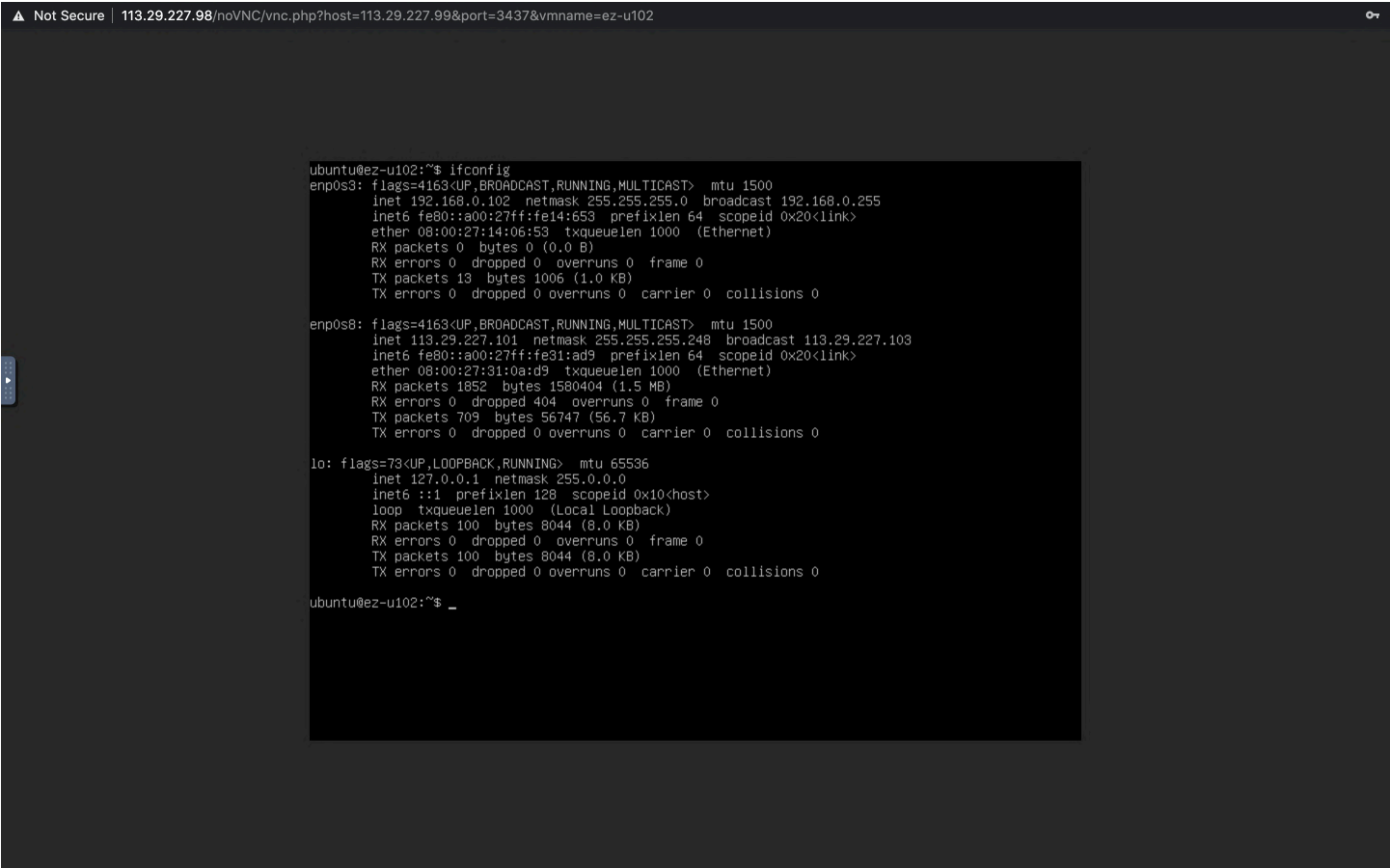
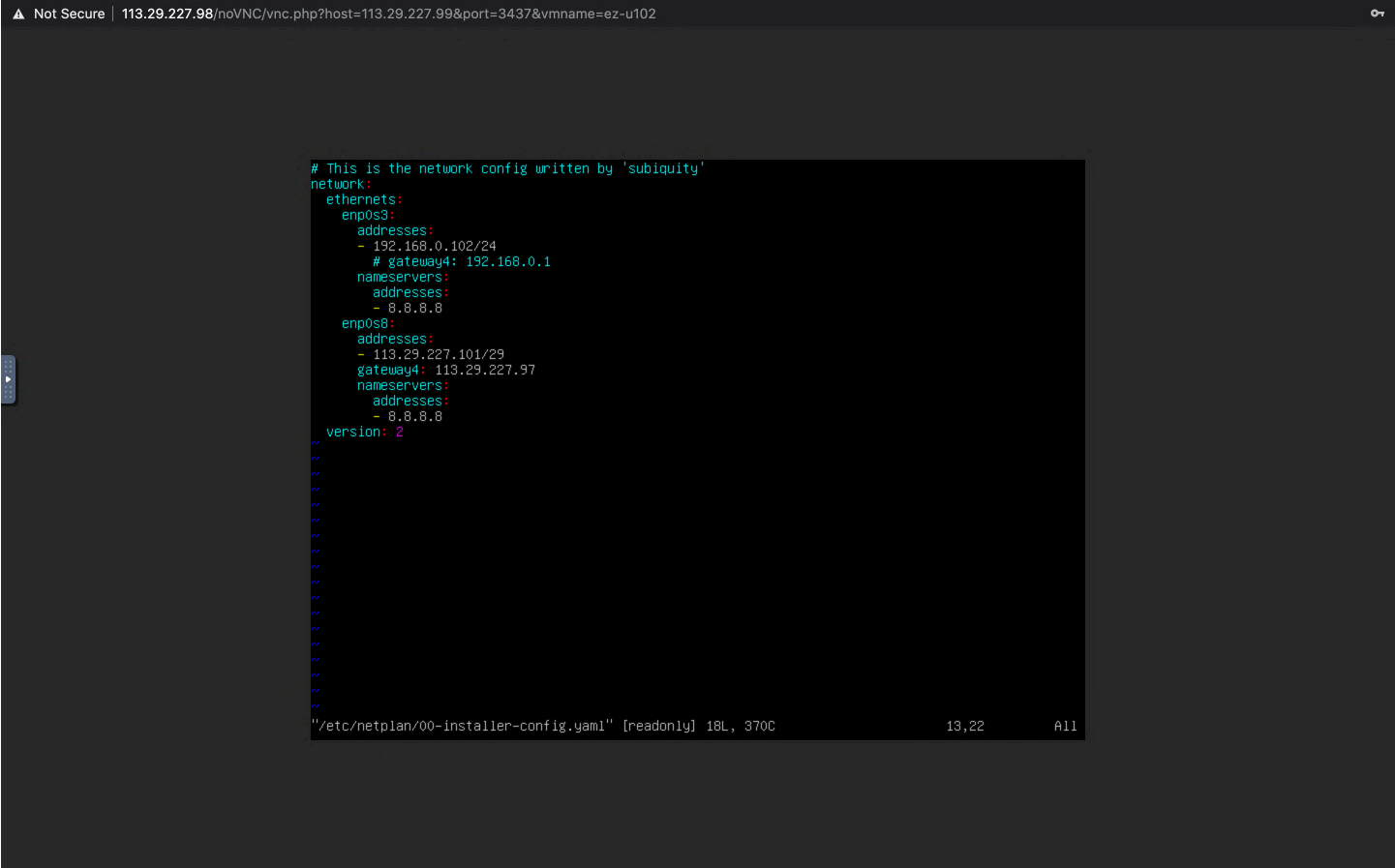
END OF TEST
This concludes a successful user replication setup and test

GUIDE #2 ADDITIONAL DOCUMENTATION: A Data Replication Scenario Where Source and Destination Are In Separate Locations / Geography

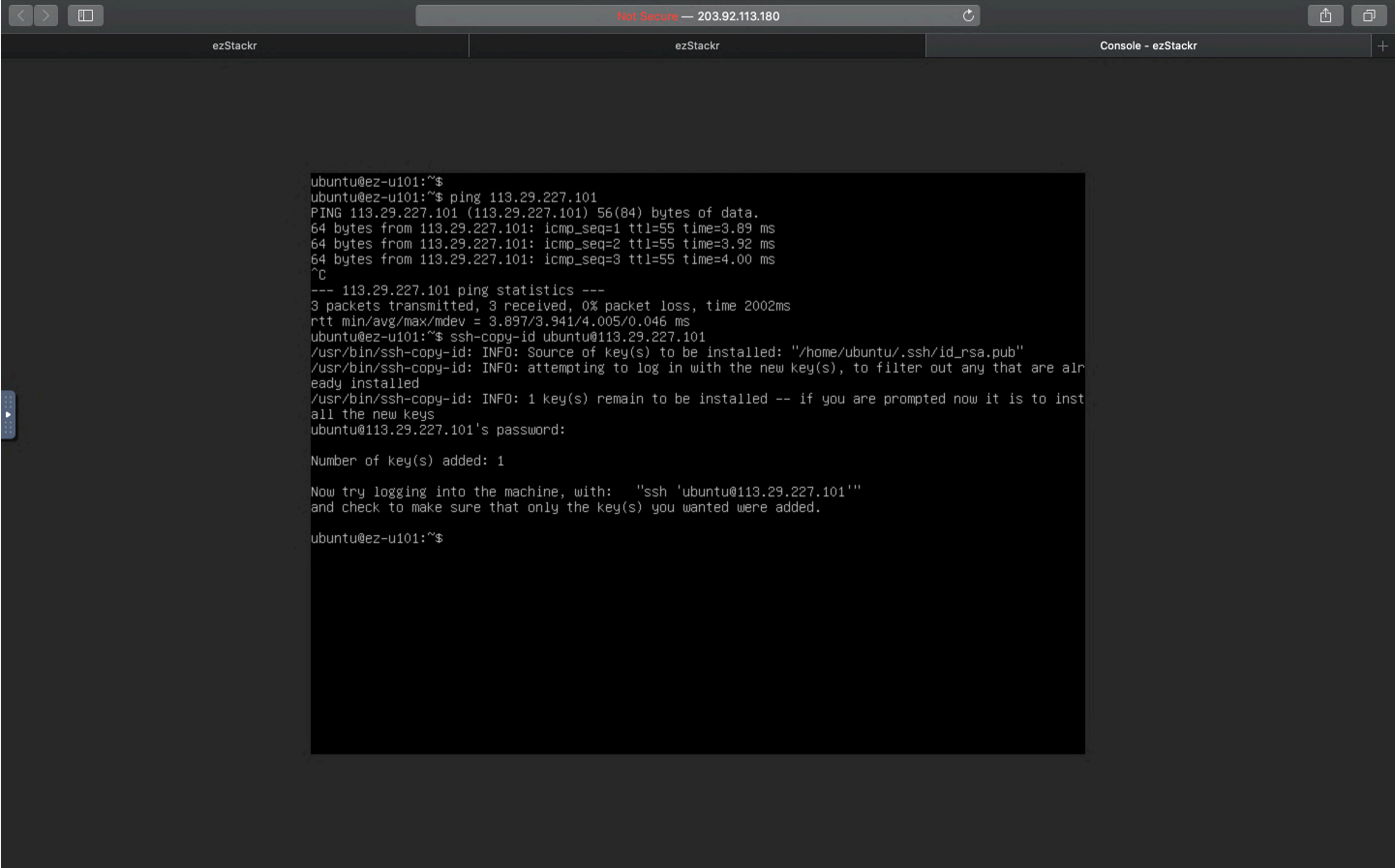
Replication Scenario:
In this test scenario, the **Source** will replicate data to the **Destination**.



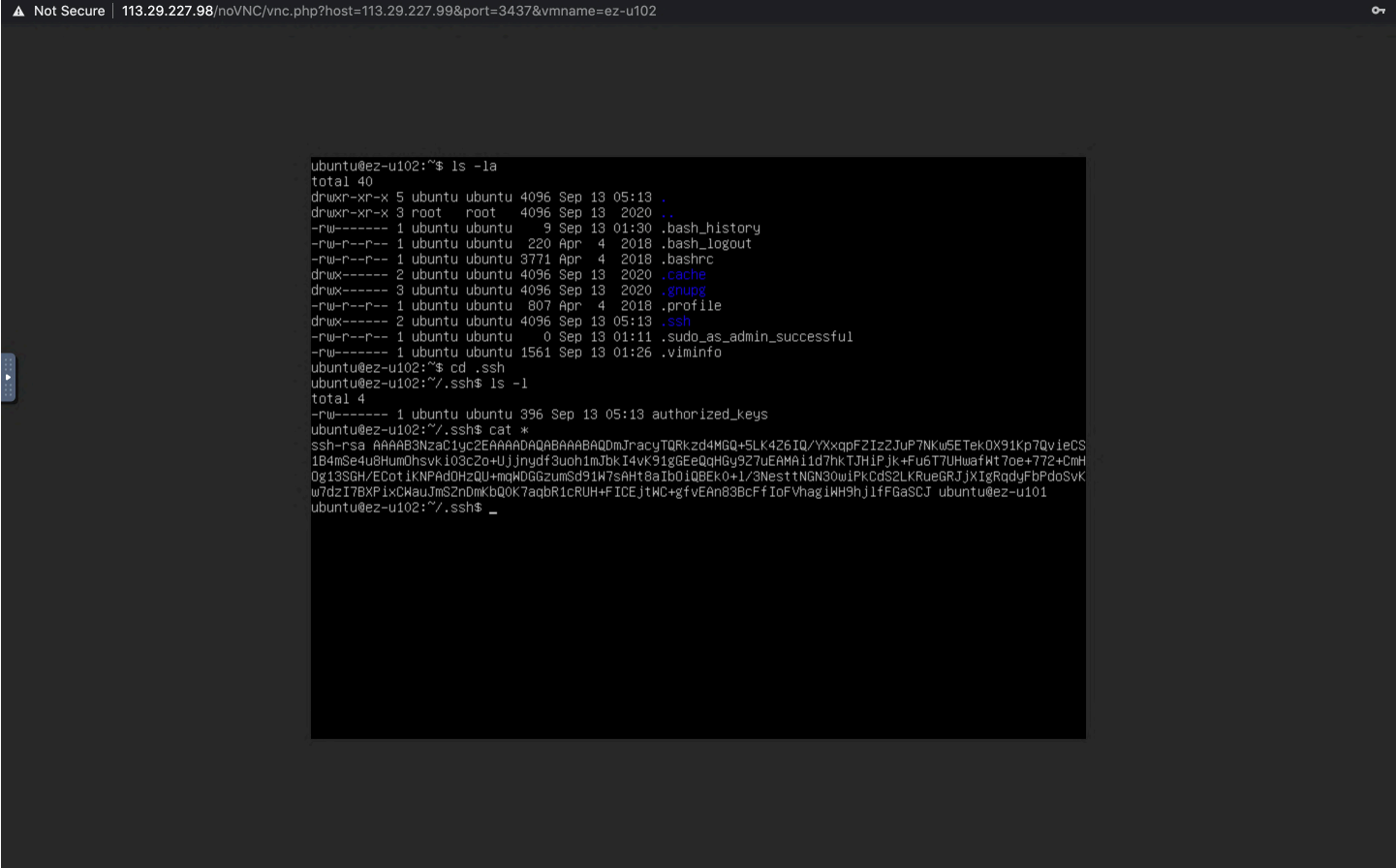
Follow the instructions from previous guide / documentation. In this example, Destination is reachable via a Public IP.



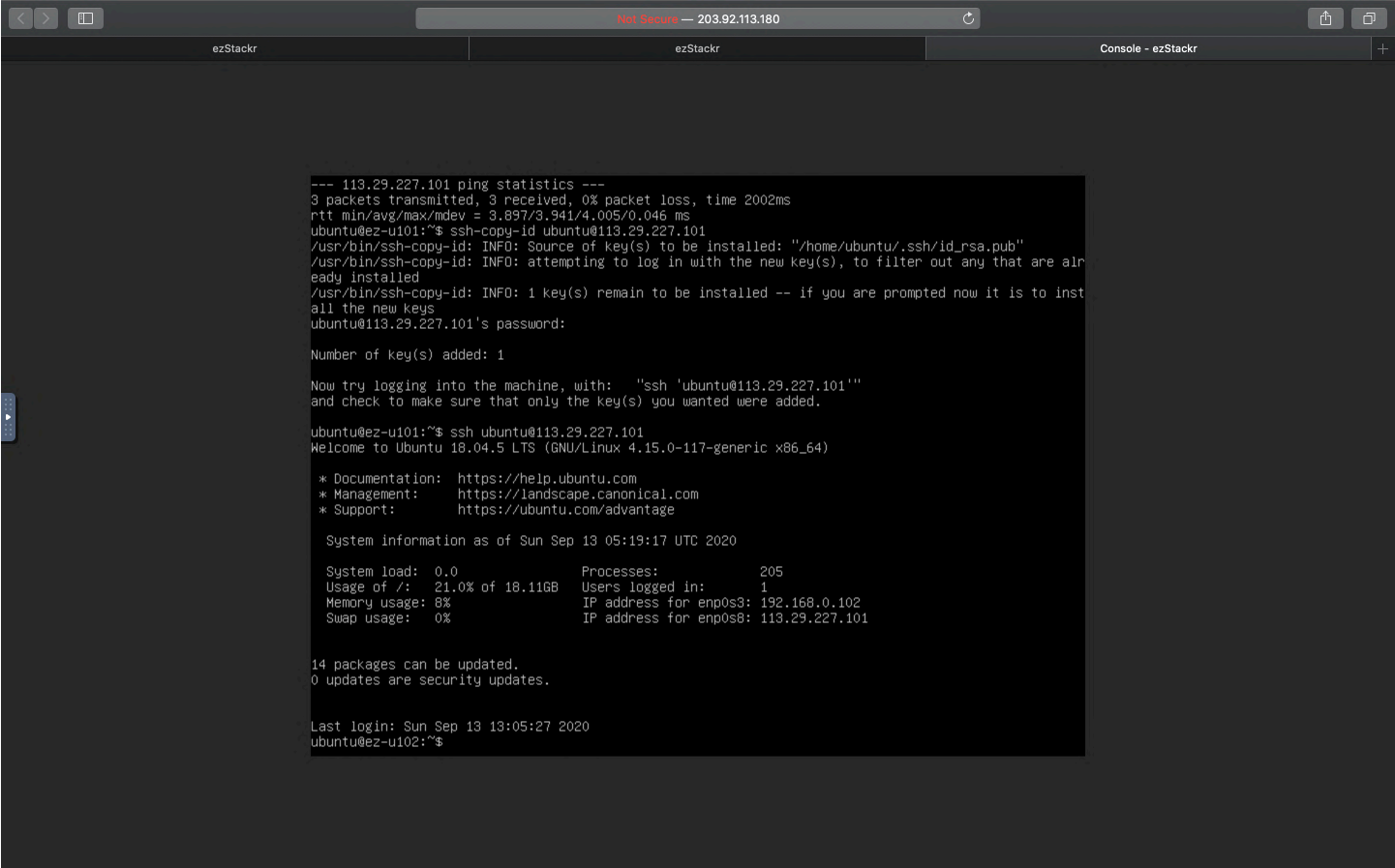
Copying Source’s Key to Destination



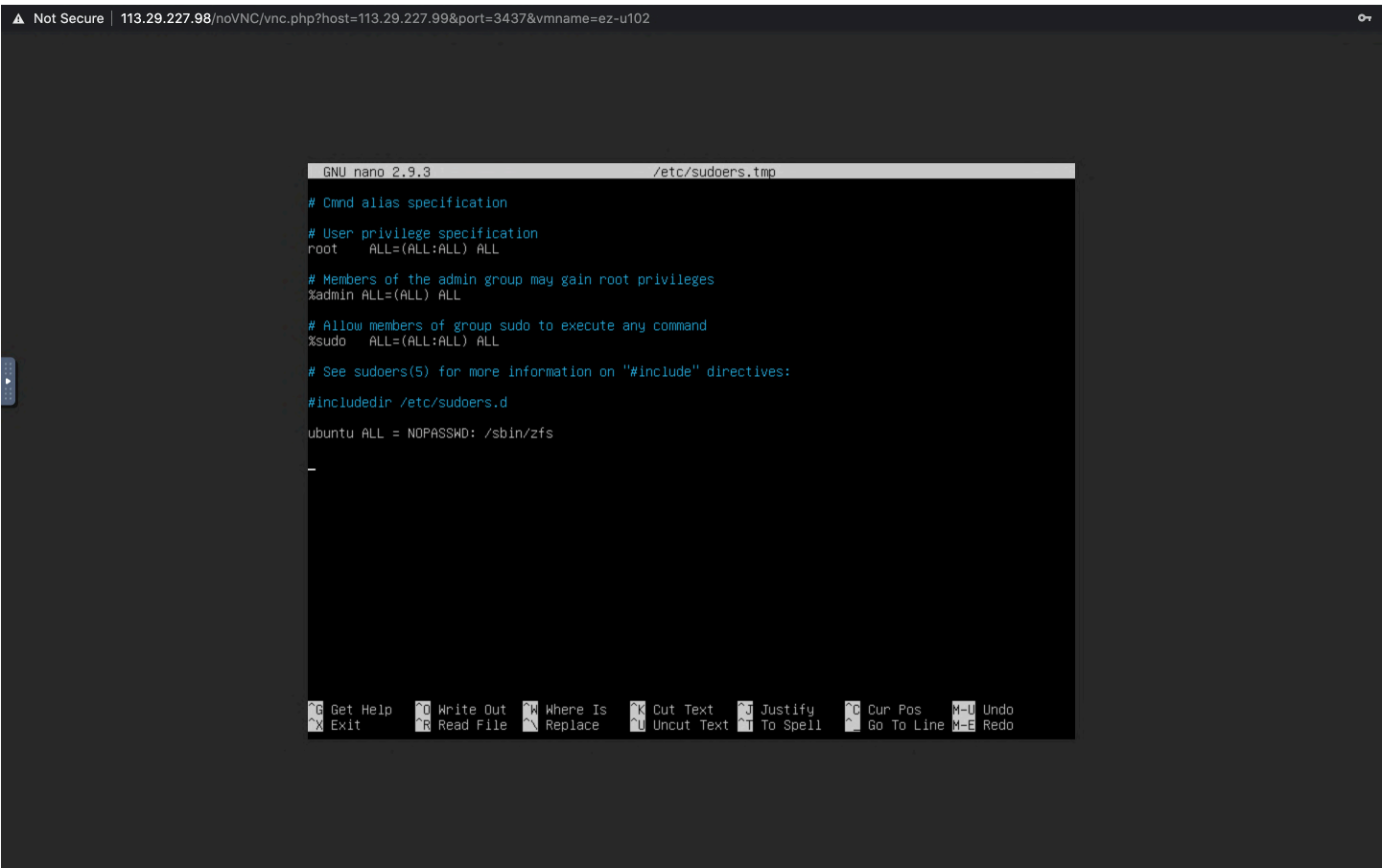
Checking that Destination has received the Source’s Key



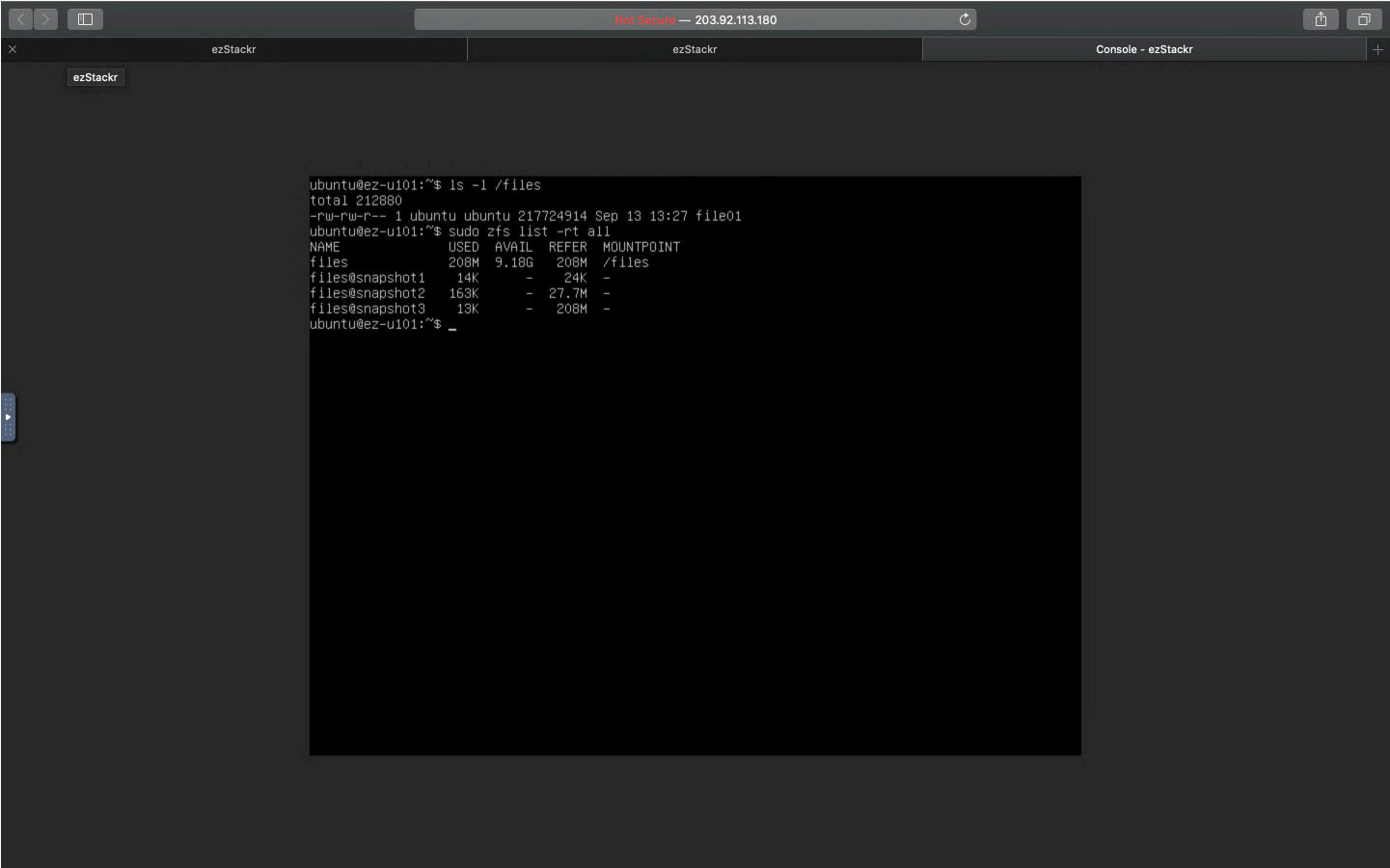
Verifying that login from Source to Destination is successfully automated



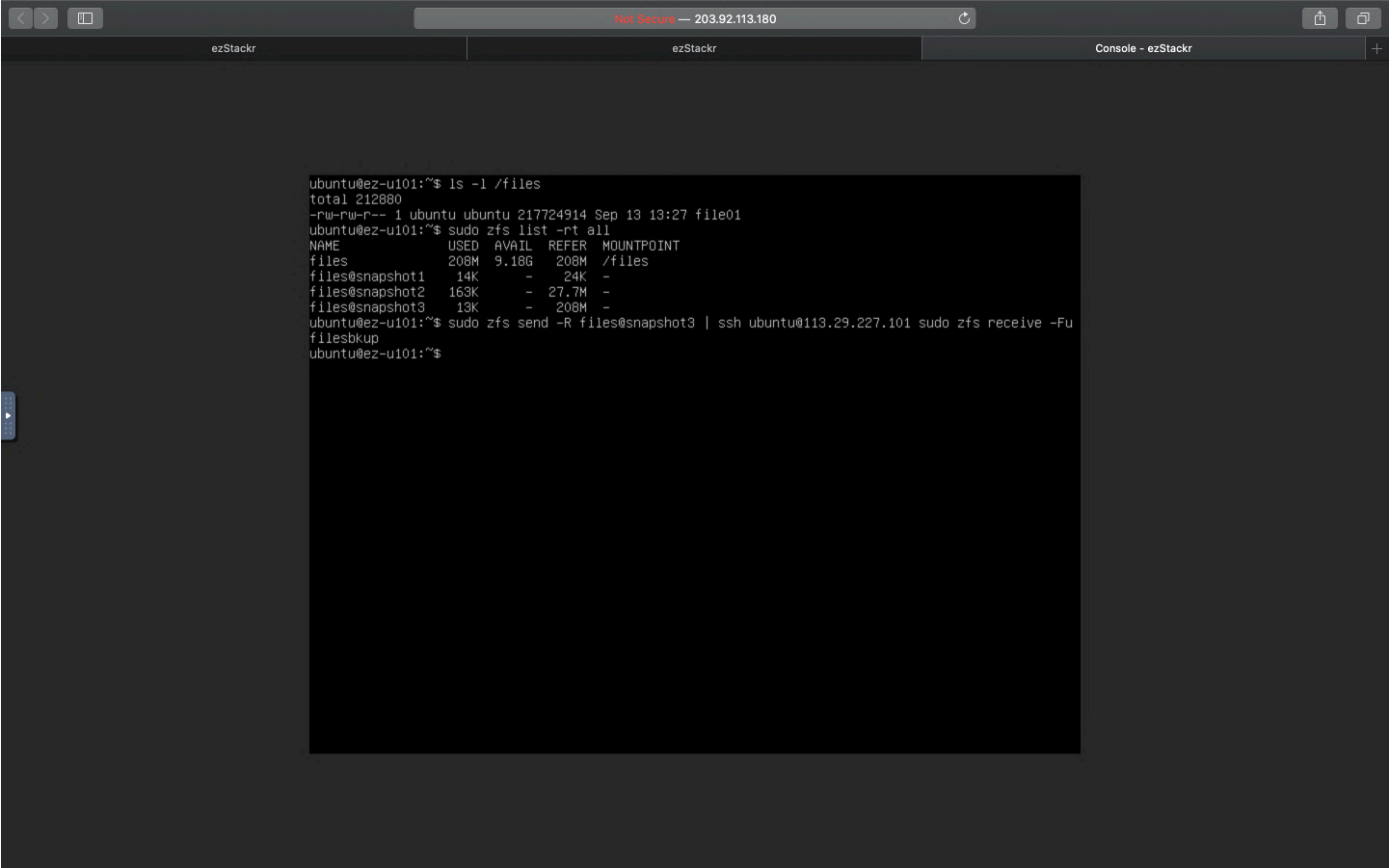
Configuring destination VM to run zfs without prompting a password when destination executes "sudo zfs". If you do not do this step, there will be an error message "no tty present and no askpass program specified". Do this at the destination VM:



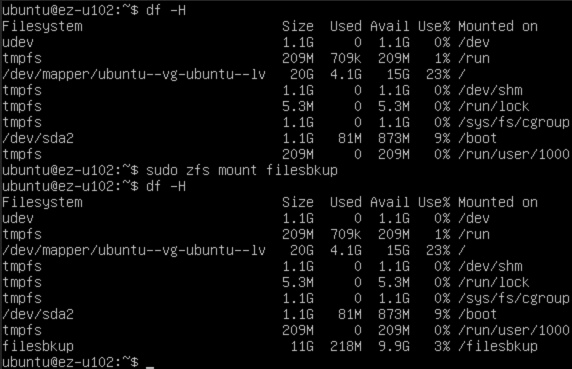
Generating a large file on Source. Taking a snapshot e.g. in this case files@snapshot3



Sending snapshot3 to Destination. About 210MB of data was received in 20 seconds.

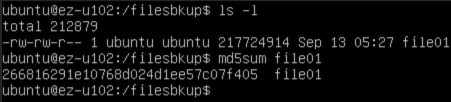


Verifying the Destination's snapshot3



```
Not Secure — 203.92.113.180
ezStackr      ezStackr      Console - ezStackr
+

ubuntu@ez-u101:/files$ ls -l
total 212880
-rw-rw-r-- 1 ubuntu ubuntu 217724914 Sep 13 13:27 file01
ubuntu@ez-u101:/files$ md5sum file01
266816291e10768d024d1ee57c07f405  file01
ubuntu@ez-u101:/files$ _
```



This concludes a successful user replication setup and test

LGA Telecom Pte Ltd
33 Ubi Avenue 3
#08-53 Vertex (Tower A)
Singapore 408868

Tel (65) 6892 2308
Email: sales@lgatelecom.net
Website: www.lgatelecom.net